Who are we?

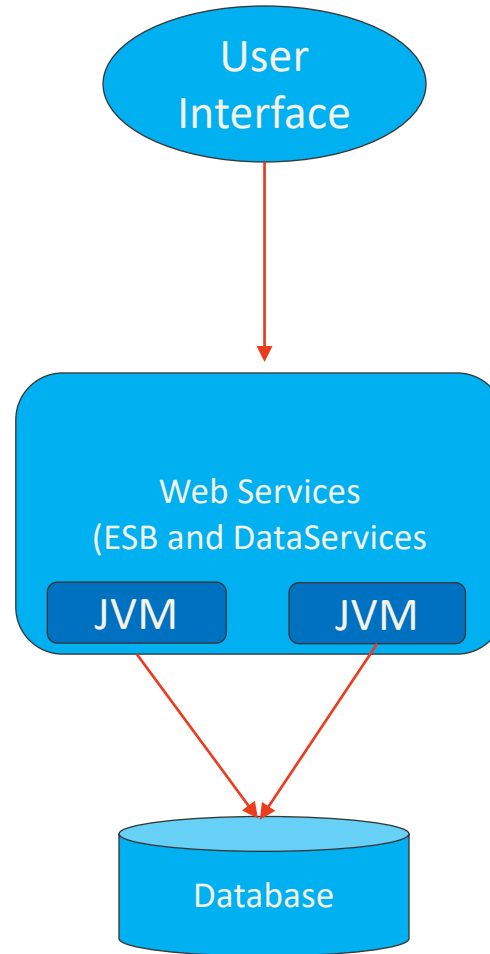24 Hour Fitness is a leading fitness industry pioneer with more than **400 clubs** across the United States. 24 Hour Fitness has **20,000 plus employees** serving our nearly **4 million** club members.
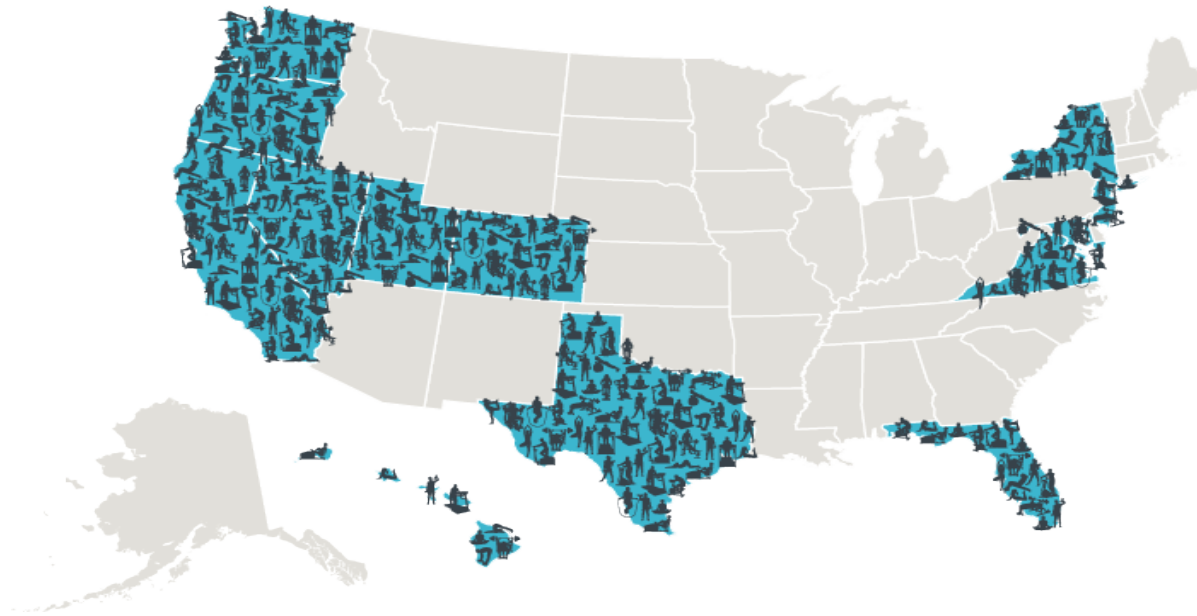
Should you use an In-Memory Computing solution?

Use-cases

Our Application Tiers

User Interface

Web Services
(ESB and DataServices

JVM   JVM

Database

Didn't I just pay my bill? How come your application says I still owe $50?

Self-Service Online Payment UI

JVM JVM

3. Pay some invoices

1. Get balance

Accounting Service

JVM JVM

4. Charge card

2. Query invoice balances

5. Save receipt

Payment Service

ERP DB

Staging Tables → Base Tables

6. Batch every 6 hours

Enter a Distributed Cache, it knows I already paid

Self-Service Online Payment UI

JVM JVM

Now we can filter out paid invoices from the get balance response.

3. Pay some invoices

1. Get balance

Accounting Service

JVM ←Distributed Cache→ JVM

What if we implement a distributed cache such that cache consistency is retained across the JVMs so we know which invoices have already been paid?

Payment Service

4. Charge card

5. Save receipt

2. Query invoice balances

ERP DB

Staging Tables

6. Batch every 6 hours
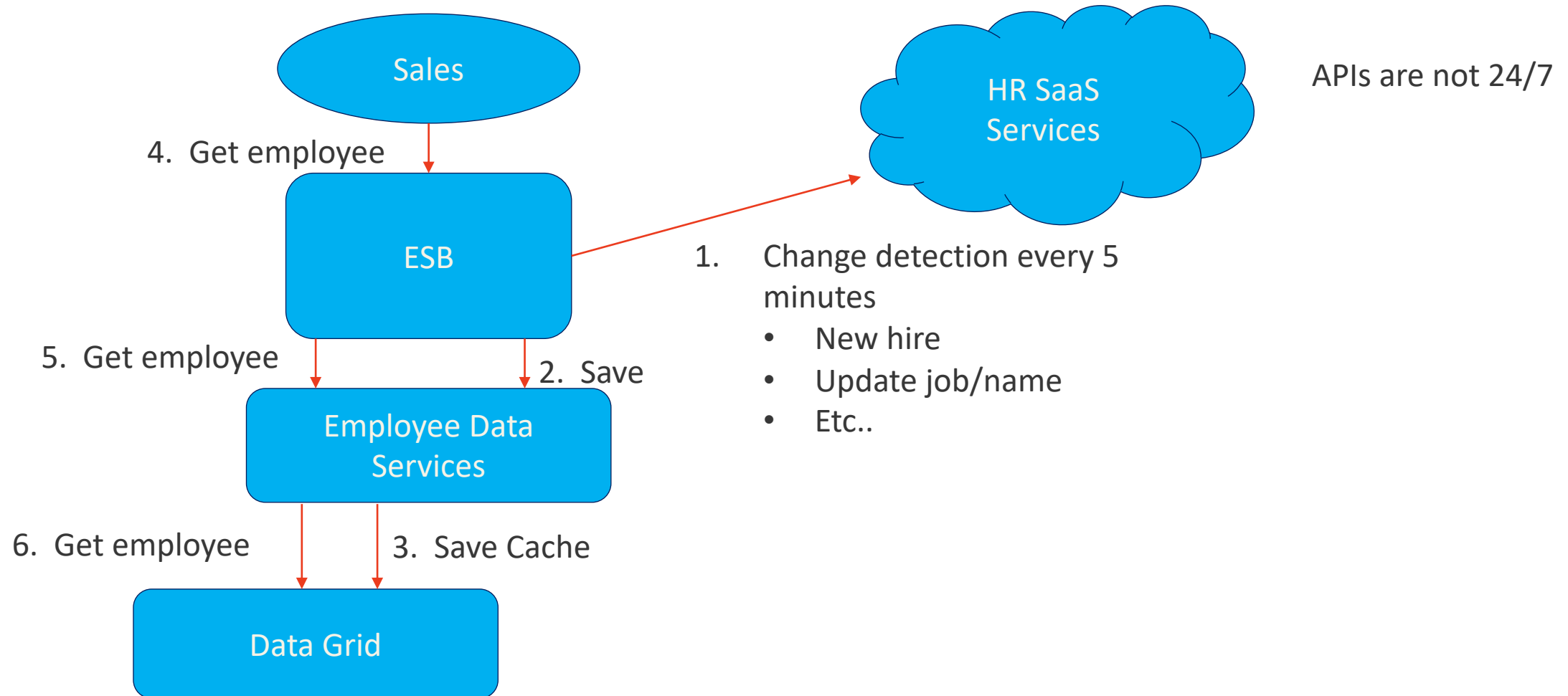
Base Tables

Should we host our HR System on-prem, or go with a Cloud solution?

Life is perfect!, no need for an on-prem database, dare I say Nirvana?

HR SaaS APIs

Sales

1. Get employee

ESB

2. Get employee
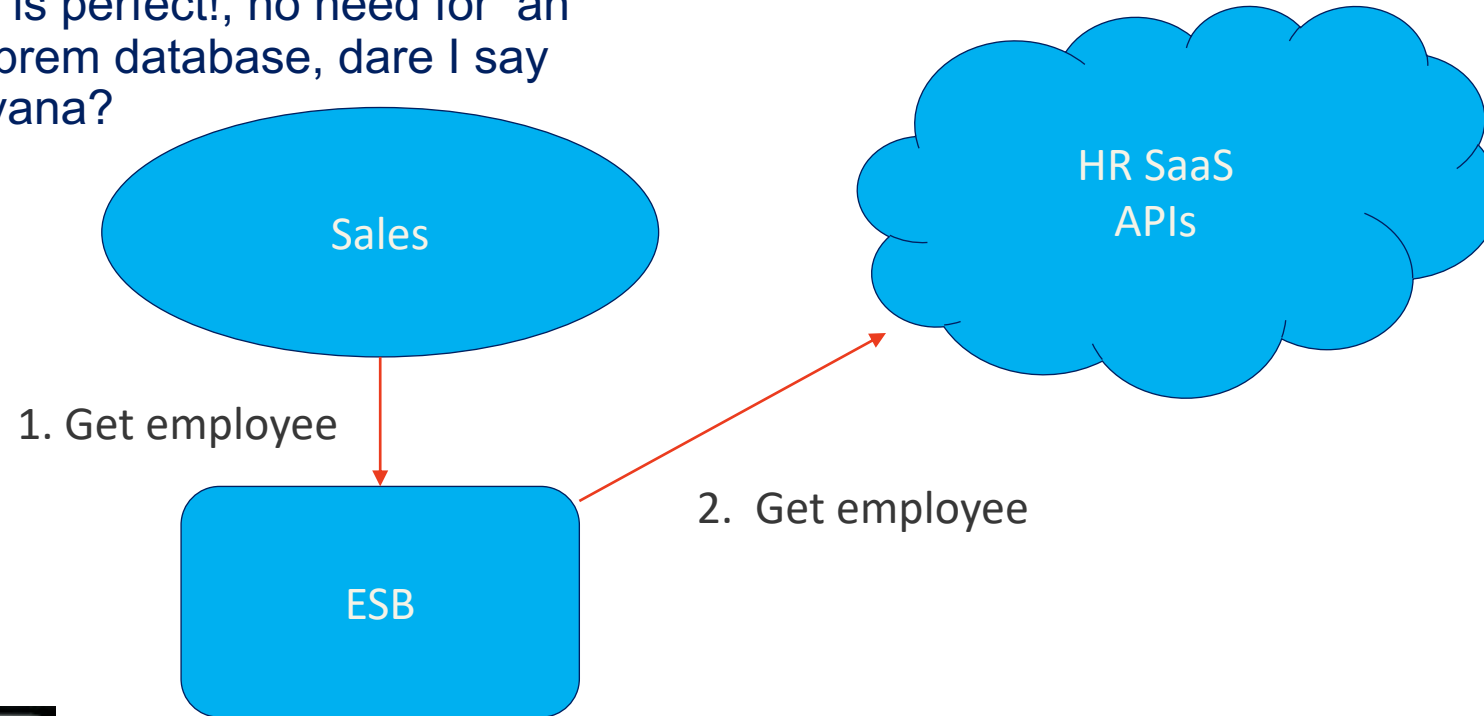
We have some work to do, we need a solution!

Sales

HR SaaS Services

APIs are not 24/7

4. Get employee

ESB

5. Get employee

2. Save

Employee Data Services

6. Get employee

3. Save Cache

Data Grid

1. Change detection every 5 minutes
    - New hire
    - Update job/name
    - Etc..

To build or buy a Lead Management System, that's the question

Sales

Cloud Lead Management Services

1. Purchase

ESB

2. Create lead
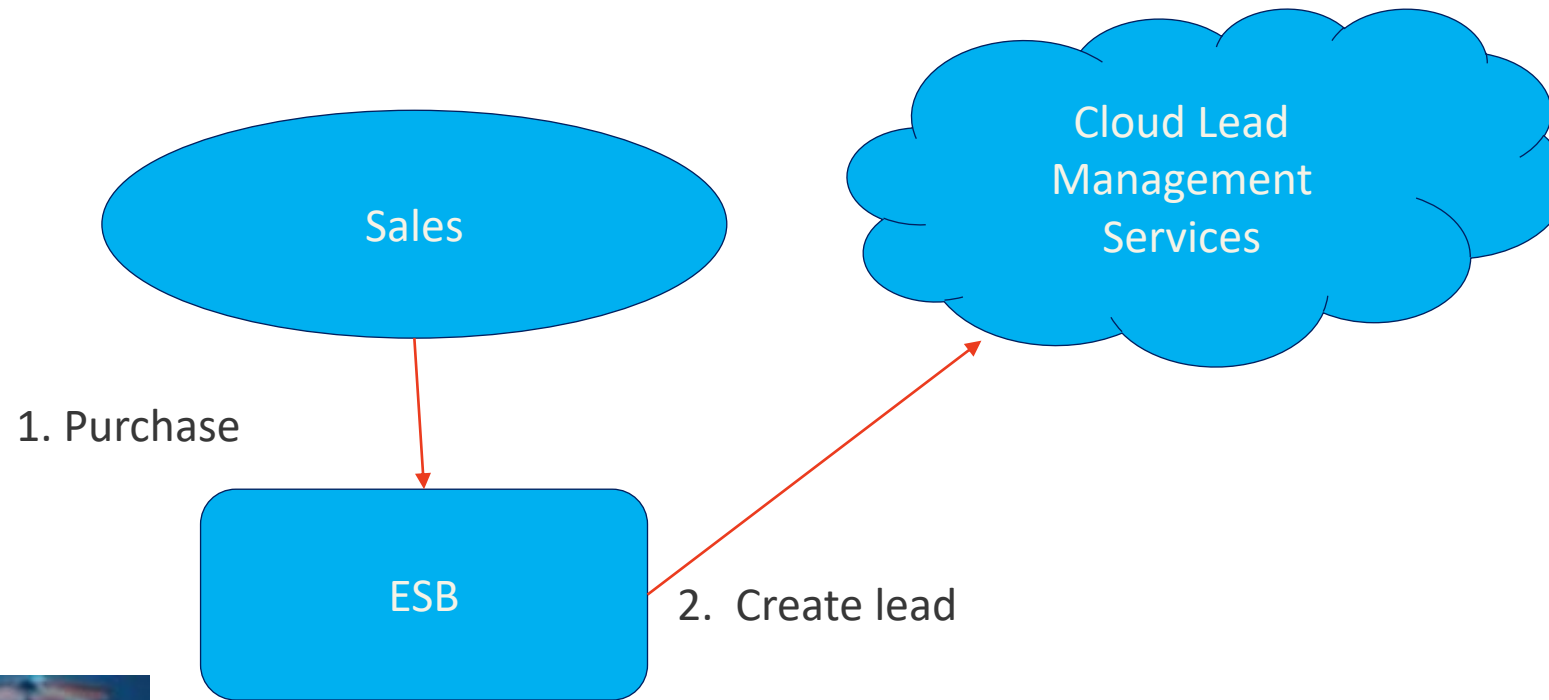
Help, we need fast, not slow and chatty!

State	GUID
CA	2c92c0f86d397f11016d4062ed631f7
OR	2c92c0f96875c0f001687daa988d1ce
...	...

Sales

Cloud Lead Management Services

1. Purchase

ESB

2. Get lead origin GUID
3. Get state GUID
4. Get club GUID
5. Create lead with GUIDs
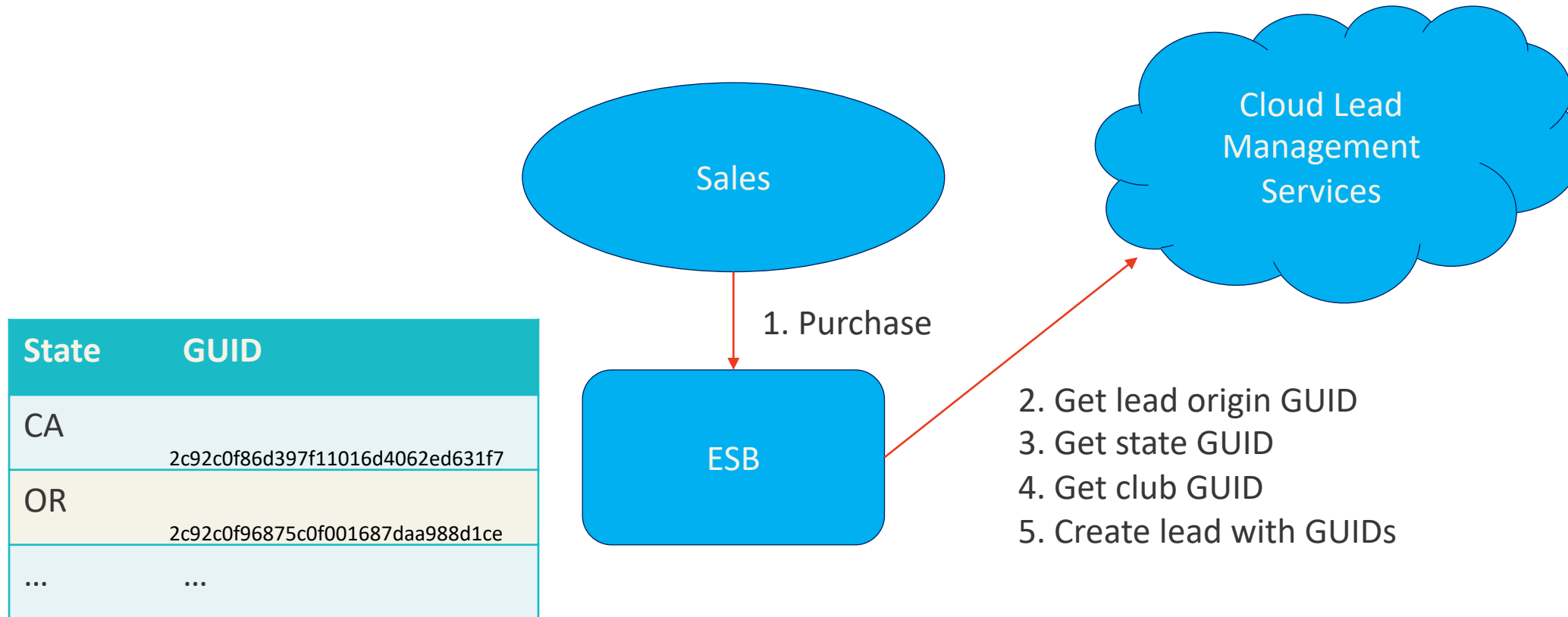
This solution is fast and less chatty

State	GUID
CA	2c92c0f86d397f11016d4062ed631f7
OR	2c92c0f96875c0f001687daa988d1ce
...	...

Sales

1. Purchase

ESB

3. Create lead with GUIDs

Cloud Lead Management Services

2. Get GUIDs

CRM Entity Data Services

Get GUIDs

Loaded at startup

Data Grid
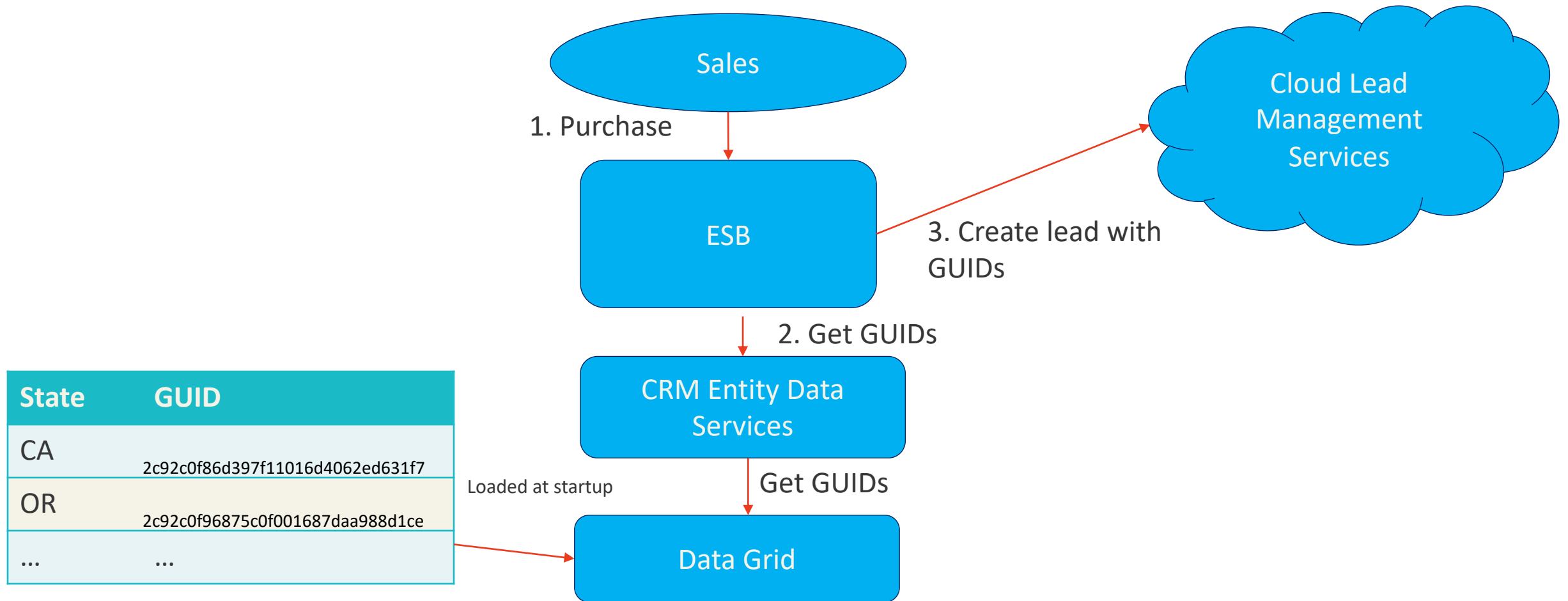
Out with the on-prem ERP
in with a SaaS billing system

Club Check-in

Sales

Member Service

1. Rate Limits
2. Can't search

2. Member Check-in

1. Purchase

3. Search

ESB

1.1 Create purchase subscription

2.1 Do they owe us $$$ and is the subscription active?

3.1 firstName like 'Cr%' and lastName like 'Gre%

SaaS Billing System
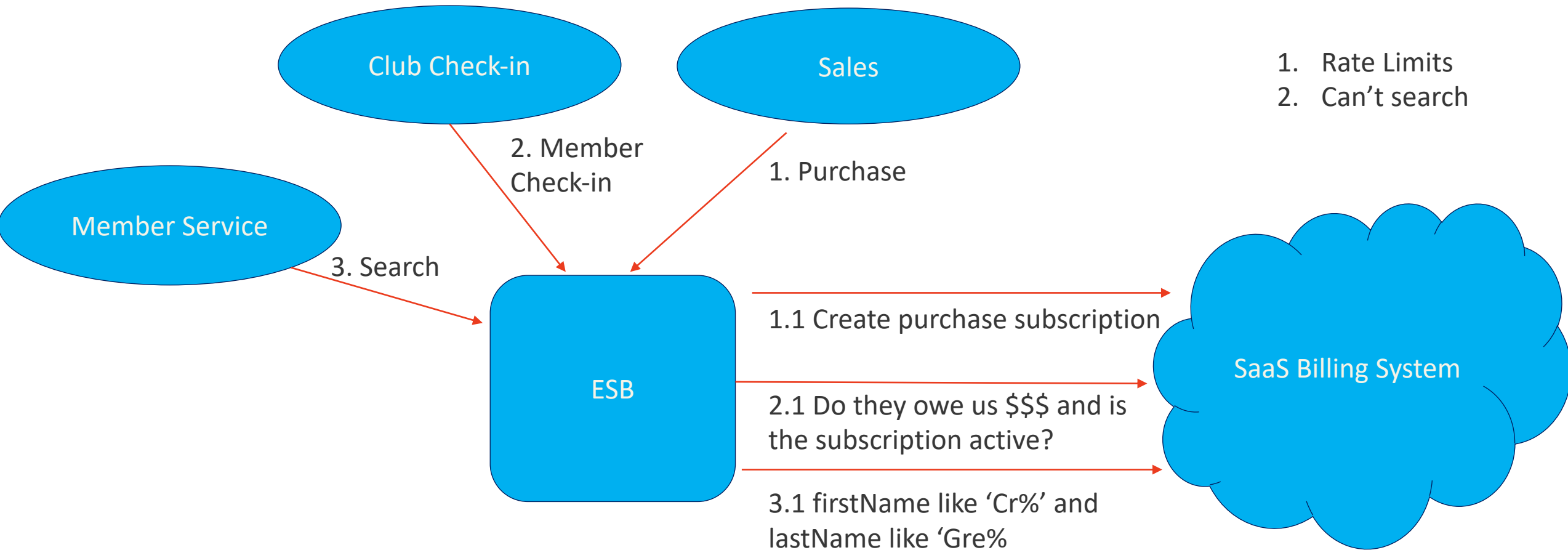
Solving API limitations with an IMDG

Member Service

Club Check-in

Sales

SaaS Billing System

3. Search

2. Member Check-in

1. Purchase

ESB and Data Services

JVM   JVM

1.1 Create purchase subscription

2.1 Do they owe us \$\$\$ and is their subscription active?

3.1 firstName like 'Cr%' and lastName like 'Gre%

Get delta data (**every 15 minutes**)

No nightly ETL, Fast

In-Memory Data Grid

Server Node   Server Node   Server Node   Server Node

JVM   Cache Loader   JVM

Load data

I came to work out!
NOT stand and wait at the front desk!

Our legacy member check-in approach:

1. Did **nightly summary of a members status** and stored it into a disk based database table
   - Extra storage (4 million records)
   - Extra processing (nightly job looking at all data)
   - Stale data is 24 Hours old
     - I paid my bill online this morning, why at 6 pm do you say I owe you $?

Our new In-Memory Data Grid approach to Member check-in:

1. Operates on **near real-time data querying The Grid**
   - As fast as we can get it from our Biling System – every 15 minutes vs. nightly

2. Is **faster** in terms of response times
   - Customers spend less time at the front desk
   - Might need less member check-in terminals in each club as lines are less likely to form

Reasons why you might use an In-Memory Data Grid

1. API limitations, some Cloud/SaaS APIs:

   - Are slow and chatty – a double edge sword

   - Are not guaranteed to be up 24/7

   - Have rate limits

   - Can't support searches (LastName like 'Gre%')

   - Can't support joins, only support single entity/object querying

   AND you desire

2. Fast, scalable, near real-time, future proof solutions

   How fast is it? Thought you'd never ask…

Volume of data

1.5 million (member) accounts

1.8 million Subscriptions

2 million Rate plans

8.2 million Rate plan charges

3 million Invoices

9 million Invoice items

1.4 million payments

It is both fast to load data, and retrieve data

I feel the need, the need for speed.

Transactions & services > datasvcs-prd Web Service: WebService/GetSubscriptions-1.0.0 > Details

▽ Filtered by | Request type: Default requests ✕ |
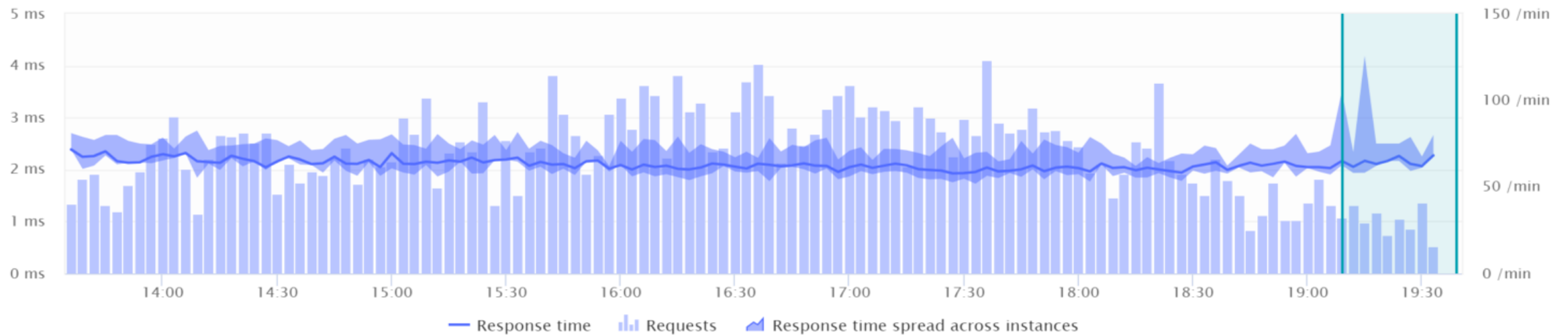
| Response time | Failure rate | CPU | Throughput |

View | Median | Slowest 10 % | Slowest 5 %

— Response time   ▫▫ Requests   ◿ Response time spread across instances
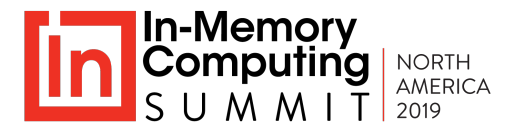
Digital Transformation

Getting ahead of the game

Digital Transformation

The API Economy
Consider solutions that future proof your company

2000

2001 - 2010

2011- 2017

2017- 2019

2020 - ?

24 Hour Fitness grapples with **selling memberships on the web**, and a commissioned sales force in the clubs! **The start?**

.doc boom

Websites and self-service

Smartphones

Mobile traffic exceeds web traffic

Smartwatches

VUIs emerge
- Alexa, Google Home

IOT emerging
- Refrigerators

More VUIs
- Cars

More IOT
- Smart homes

What's Next?
- Rate of change

The 3 V's and data

Is it your turn!

Do you see any potential **In-Memory use cases** at your company?

Are you a visionary at your company?

Why not make it happen?

You might need to:
1. **Evangelize and socialize** your solution over months or years
2. Do **vendor/product evaluations** and negotiations
3. **Own the products and solutions** you evangelize
   - In good times and bad
4. **Step out of your comfort zone!**

Thank You and good luck on your journey!

A ship in harbor is safe, but that is not what ships are built for.
— John A. Shedd

Appendix – Useful links

My 2019 IMC Summit "Tales from the trenches" presentation

- https://www.imcsummit.org/2019/us/session/gridgain-ultimate-edition-aids-implementation-saas-systems-and-replaces-traditional

Data: Volume, Variety, Velocity

- https://whatis.techtarget.com/definition/3Vs