

RED HAT FORUM, SEPTEMBER 2018

SAVING THE DANISH WELFARE SYSTEM USING OPENSSHIFT

Date: September 25, 2018

Version: 1.0

Author: Michael Vognsen Nielsen & Dan Strang Søndergaard

Contact: mvn@netcompany.com & dss@netcompany.com

netcompany



© Copyright 2018 Netcompany. All rights reserved.

Agenda

- About us
- Introduction to Netcompany
- SKAT ICI Project Background
- Demo – Ordering new environment
- Introduction of ICI
- Configuration management and Continuous Integration on ICI
- OpenShift in ICI – what happens when we instantiate a new environment?
- Benefits and lessons learned
- Demo – Show new environment in OpenShift
- Q&A

INTRODUCTION TO NETCOMPANY

LOCATIONS AND EMPLOYEES

Netcompany is the fastest growing and most successful IT services company in the Nordics, aiming to become a Northern European market leader.

We are committed to lead the way showing how digital transformation can create strong, sustainable societies, successful companies and better lives for all of us.

Netcompany was founded in 2000.

1.4bn
Revenue (DKK) in 2017

+20%
Average growth for +10 years

5
Countries of operation

+1800
No. of employees



FULL STACK IT SERVICES PROVIDER

Leading in all areas
of the stack

Digital Platforms

- Digital front end
 - CRM
 - Integration
- AI/Data analytics
 - IoT

Core systems

- Public welfare systems
 - Billing
 - Modern ERP

Infrastructure

- Cyber security
- Cloud Migration
- DevOps setup

With expertise in all
modern technologies...



open source



Java



Throughout your
digital journey...



Innovation



Development



Operation

ICI PROJECT BACKGROUND

ICI | Cleaning up after the biggest IT scandal in Danish history!

- In 2005 SKAT initiates a project to consolidate all public debt collection in one system called "EFI"
- The project was scheduled to GO-LIVE in 2007, but the project was six(!) years late and thus it was not until 2013 it went live
- However, EFI was ridden with errors and was not able to perform debt collection according to Danish law. Consequently, the system was shut-down permanently two years after go live
- At that point the amount of unpaid debt had risen to 100 BILLION DKK
 - Equivalent to around 200 million barrels of oil 😊
- This was the background when ICI (Implementeringscenter for Inddrivelse) was born

ICI | Cleaning up after the biggest IT scandal in Danish history!

- Learning from past mistakes, ICI wanted to avoid extensive requirement specifications and waterfall-approach
- Through a tendering process centered around a POC, Netcompany was chosen as vendor to deliver the new ICI-solution in the summer of 2016
- The first release went live in May 2017, and Denmark could start collecting debt again
 - We are not done
 - More public authorities are being onboarded to accelerate debt collection
- We have 150 full time consultants working on the project & with SKAT and other vendors the project is 300 people
- We use SAFe™ to manage and govern the project
 - 15 agile, cross functional teams
- All of this makes this project particular demanding in terms of the underlying IT platform

DEMO – PROVISIONING A NEW ENVIRONMENT



Introduction of the ICI project

- ICI is based on a Oracle-stack with Oracle's "Public Sector Revenue Management" (PSRM) at it's center, and a bunch of other Java applications supporting it (mainly custom-made).
- 1.0 went live in May 2017 (Netcompany was the sole supplier at the time, and had the delivery-responsibility).
 - Later we released 1.1, 2.0, 2.1, 2.2, 3.0 and 3.1 (here in August).



"Maybe in a sense we're reducing debt."

President Donald Trump

CONFIGURATION MANAGEMENT AND DEVELOPMENT OF PSRM



Configuration Management on ICI - Intro

- Big project (50-70 developers in PSRM at peak)
- About 50 people working on custom Java applications
- Traceability and versioning is a must, of course.
- All processes regarding versioning and release (except Acceptance tasks) must be automated. We want Continuous Delivery.





PSRM Configurations

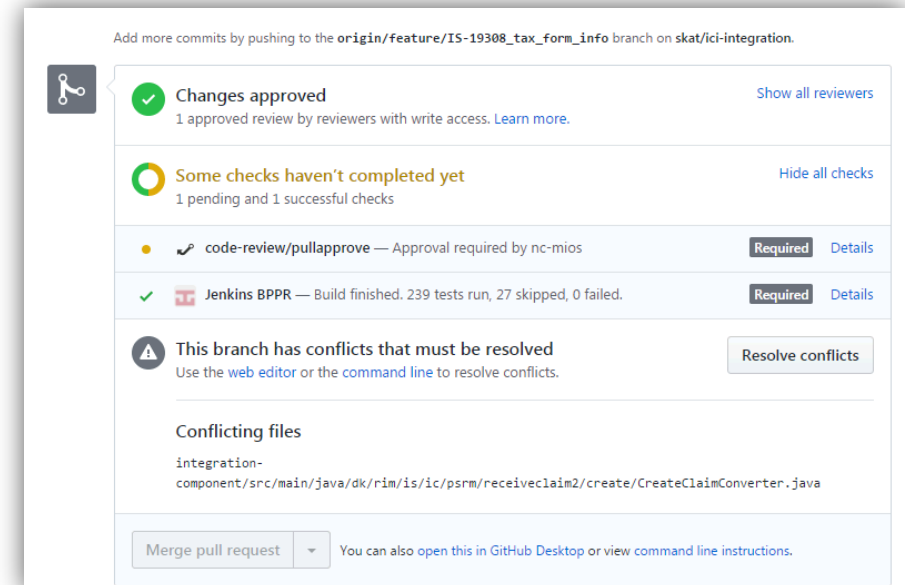
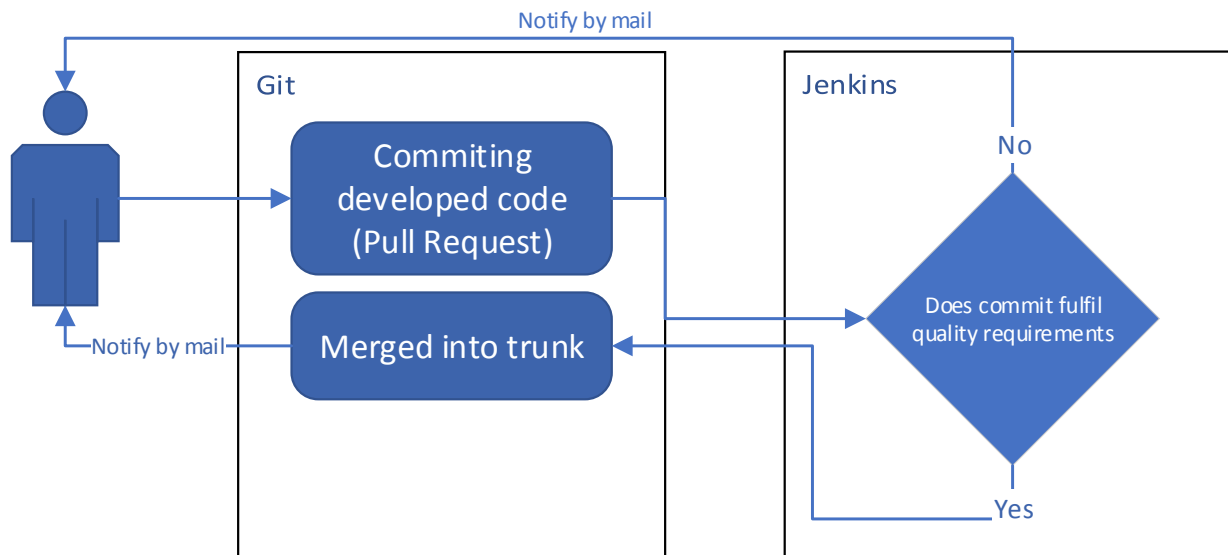
- PSRM Development is done in two ways
 - PSRM Configuration – Online scripting and mapping of objects (Maintenance Objects)
 - Java development
- There were no suitable tool for moving PSRM Configurations between environments while maintaining traceability and versioning of code. We needed a code artifact.
- We created a new tool to handle this. It checks for objects that the user have worked on, and then extracts the data directly from the development-environment's database. They are then serialized as YAML files.
- We then have a framework that translates these files into environment-independant database-injectable files used at deploy-time (or when building docker-image for DB). These can be considered our build artifacts thus are versioned (We use regular semantic versioning).
- The tool could be used for many projects based on Oracle Utility Application Framework
- Includes other less-generic features that we use in the project – blacklisting for some objects, for instance, to avoid having people commit unwanted transactional code.

```
[VAL-11133] chngd zone so payouts arent included in non-placed payments #1453
nc-har wants to merge 1 commit into develop from feat/ncp-2018-06-26-12-43-338
Conversation | Commits | Checks | Files changed | Diff settings | Review changes
Changes from all commits | Jump to... | +7 | Files changed |
10 Config/PSRM/CONTENT_ZONE/dk_gettexas_29.yml
@@ -8,14 +8,14 @@ primaryTable:
 0 ZONE_HOL_CD: F3-DE-QUERY
 9 APP_SWC_ID: F1-DEITS
104 ZONE_INITFLD: FULL
11 - VERSION: 30
12 OWNER_FLG: CH
13 chlistTables:
14 + tableName: CI_ZONE_L
15 tableKey:
16 - ZONE_CD: DK-GETTEXAM
17 LANGUAGE_CD: DK
18 - DESCR: Return Flag if exist 001 Type DNO99 Bal c08
19 VERSION: 1
20 OWNER_FLG: CH
21 DESCRLONG: ' '
@@ -209,7 +209,11 @@ chlistTables:
260 200N (SELECT * FROM CI_PAY WHERE PAY_STATUS_FLG <> 60 ) P ON P5.PAY_ID = P.PAY_ID
261 WHERE
262 SA.ACCT_ID = 'F1
263 - VERSION: 17
264 OWNER_FLG: CH
265 ZONE_PABN_VAL_OVRD: ' '
266 - ZONE_CD: DK-GETTEXAM
267 ZONE_HOL_CD: F3-DE-QUERY
268 APP_SWC_ID: F1-DEITS
269 ZONE_INITFLD: FULL
270 OWNER_FLG: CH
271 chlistTables:
272 + tableName: CI_ZONE_L
273 tableKey:
274 - ZONE_CD: DK-GETTEXAM
275 LANGUAGE_CD: DK
276 - DESCR: Get all payment events on debtor excess amount 001
277 VERSION: 1
278 OWNER_FLG: CH
279 DESCRLONG: ' '
272 + AND exists (select '1'
273 + from CI_PAY_OHAR cha
274 + where cha.CHAR_TYPE_CD = 'DK_OUTDB'
275 + and P.PAY_ID = cha.PAY_ID)
276 - VERSION: 18
277 OWNER_FLG: CH
278 ZONE_PABN_VAL_OVRD: ' '
279 - ZONE_CD: DK-GETTEXAM
```



Automated quality gates: Pull Requests on Github repositories

- All repositories is protected by direct commits. Developers are forced to use Pull Requests
- Pull Requests are qualityassured by automated tests and by manual codereview from a colleague
- Includes high-level UI tests in some repositories



WHY OPENSIFT IN ICI?

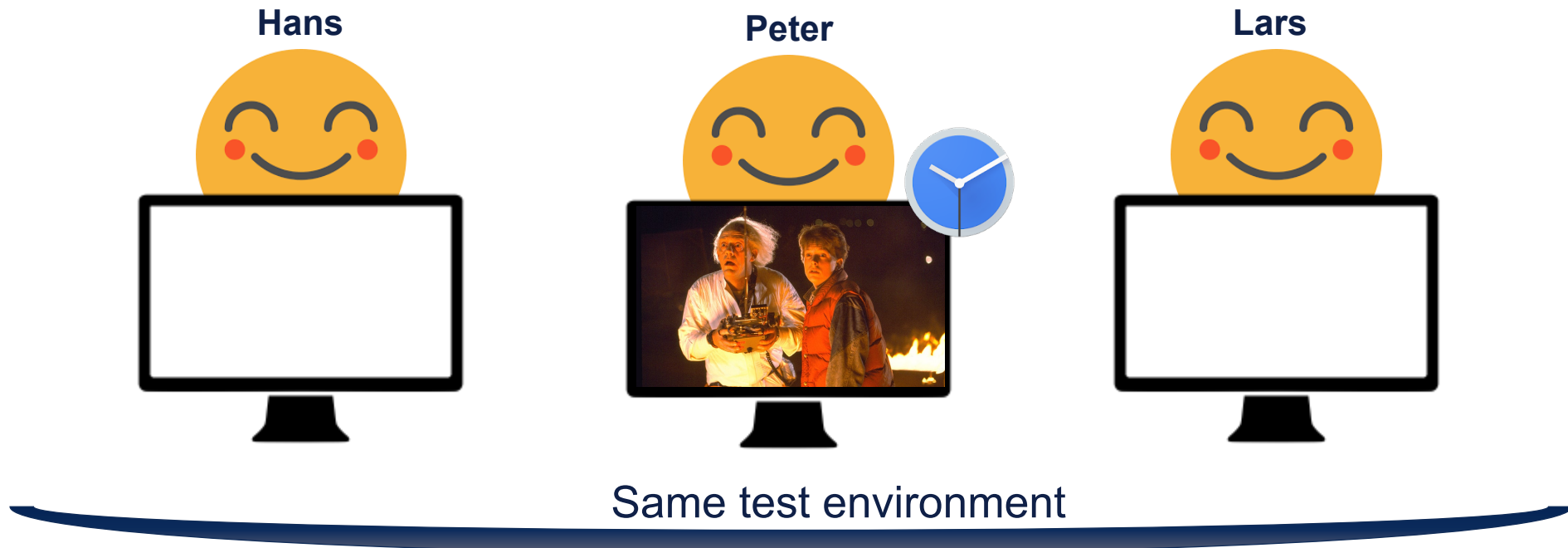
—

Cost-effective to implement applications in containers from the very beginning
(we didn't)



Why OpenShift in ICI?

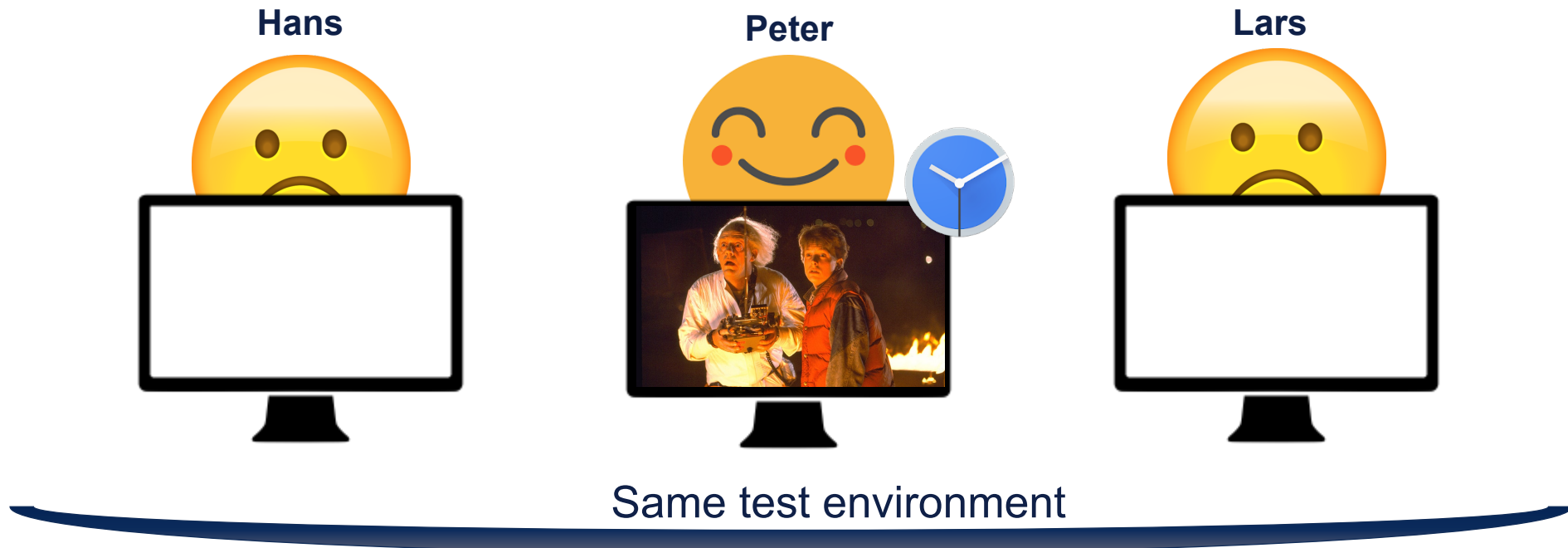
- **Test**
 - Verification (ordinary. Test)
 - Validation (Making sure that the system is legal)





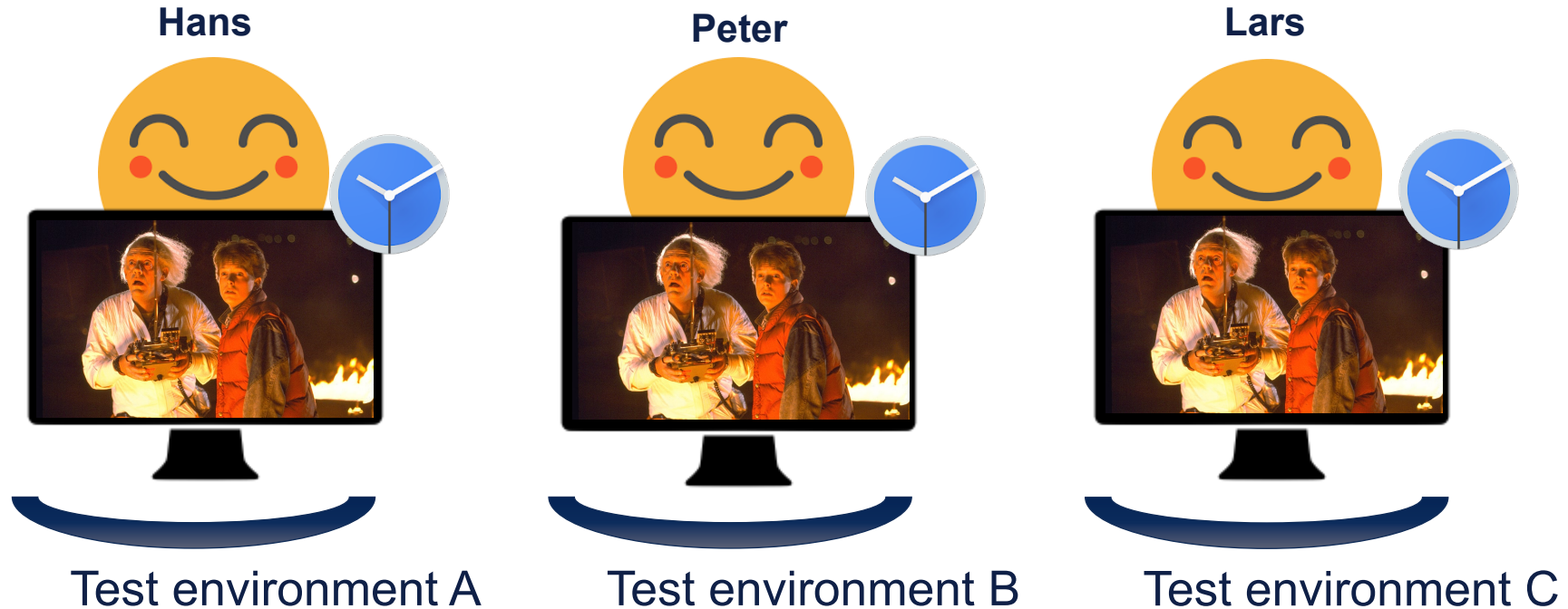
Why OpenShift in ICI?

- **Test**
 - Verification (ordinary. Test)
 - Validation (Making sure that the system is legal)



Why OpenShift in ICI?

- Test
 - Verification (ordinary. Test)
 - Validation (Making sure that the system is legal)



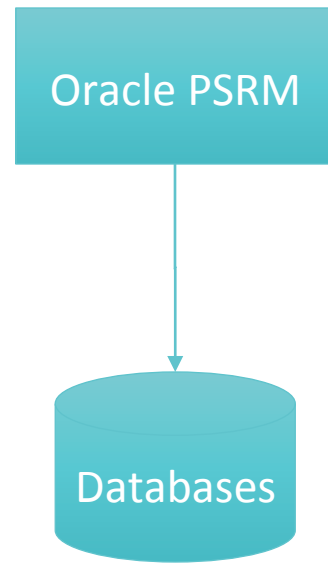


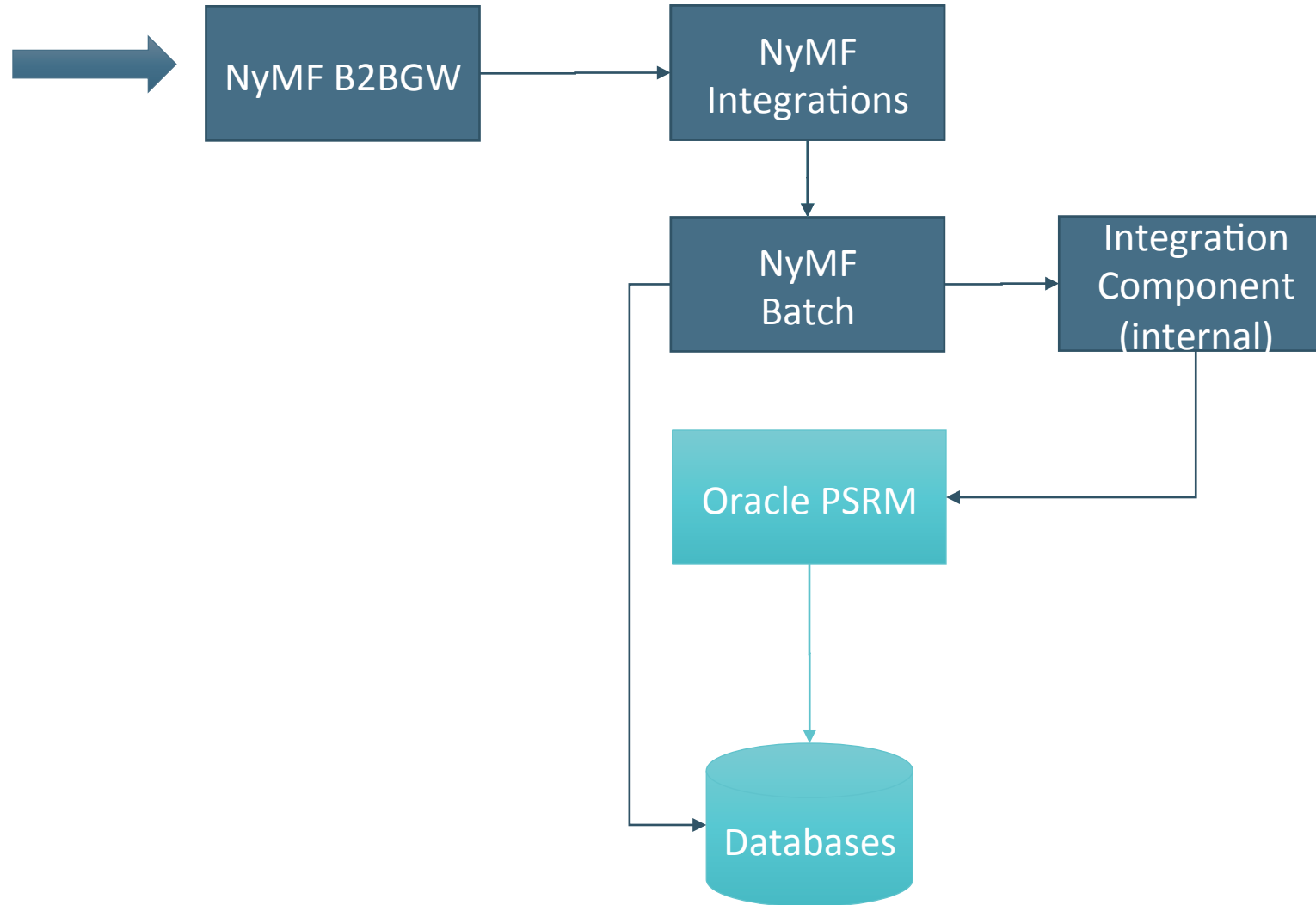
Why OpenShift in ICI?

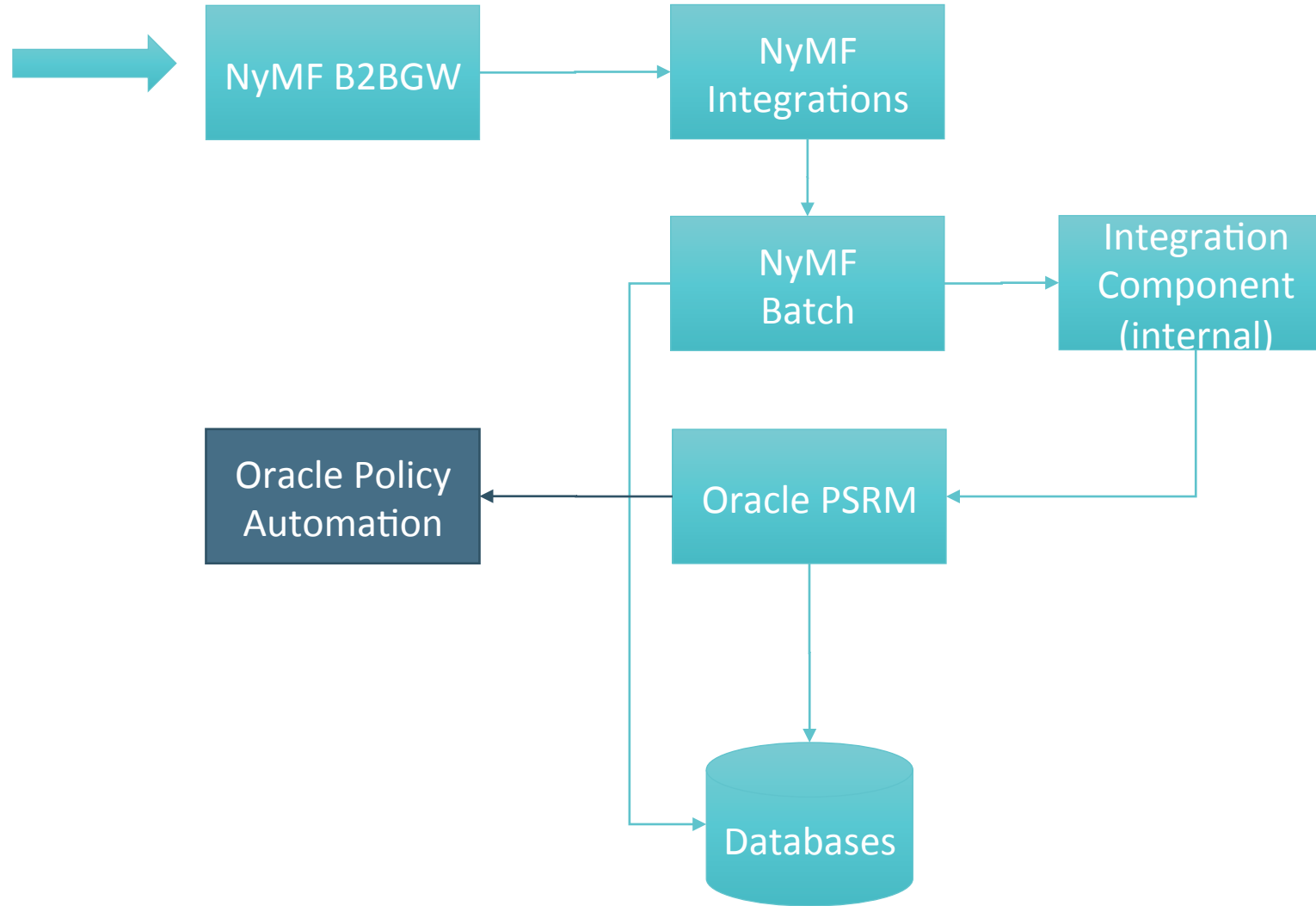
- **Not everyone was happy**
 - A lot of testers = Many environments, and development still needed to be supported.
- **Other requirements**
 - BPPR took a long time on dedicated Regressiontest environments
 - Every team needed X number of environments
 - Almost everything was already automated, but it still took a long time to order ned virtual servers, OS installation, Middleware etc.
 - A lot of people have access to environments. Middleware was tweaked from time to time
 - Enviroment-dependant errors (time sink)
 - Used a lot of time on debugging and supporting the solution with a growing amount of environments.
 - Last fall we had 35 environments on "classical infrastructure" – and the Program was screaming for more environments
 - We committed us to OpenShift & Docker in november 2017

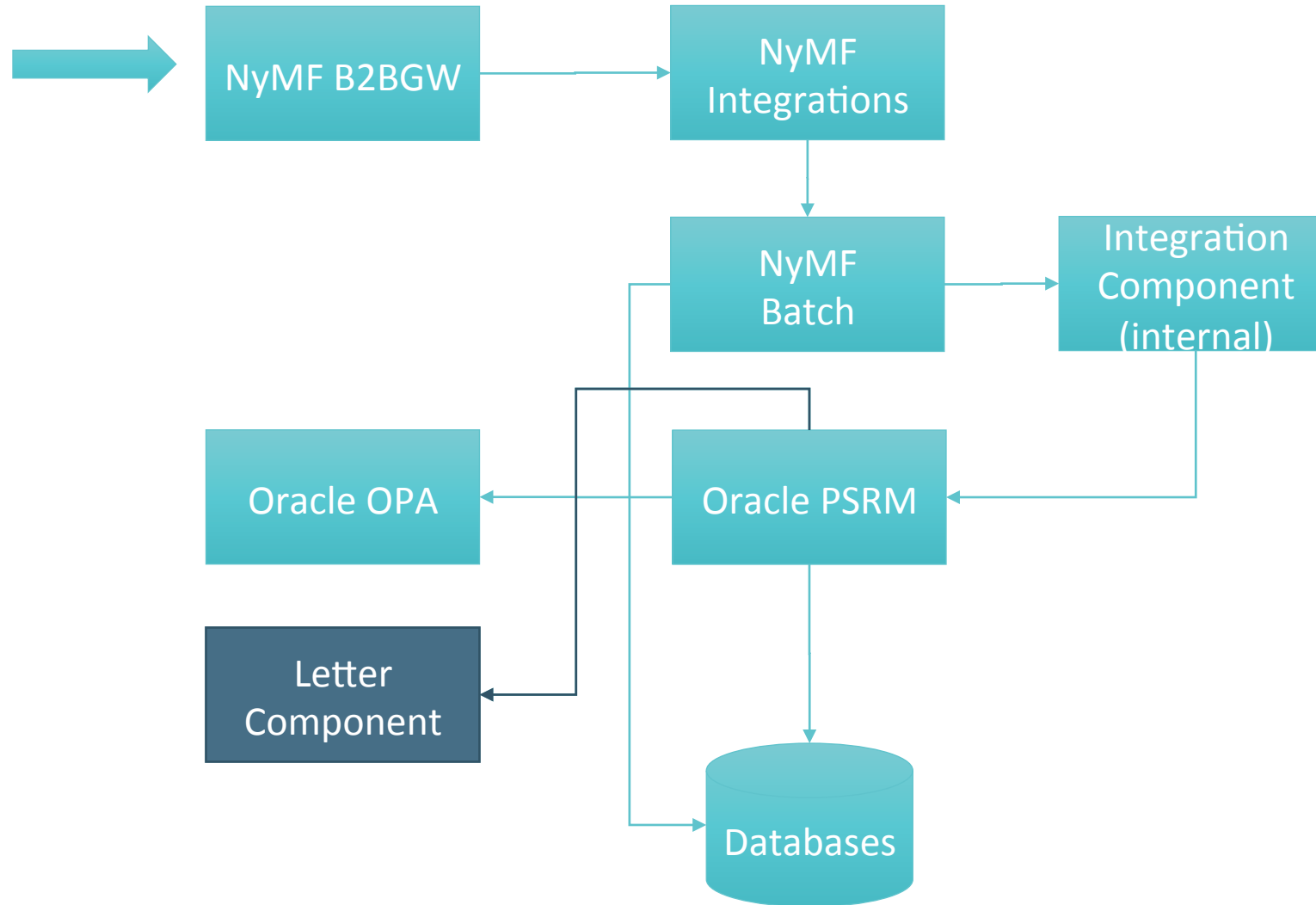
OPENSIFT IN ICI – WHAT HAPPENS WHEN WE PROVISION A NEW ENVIRONMENT?

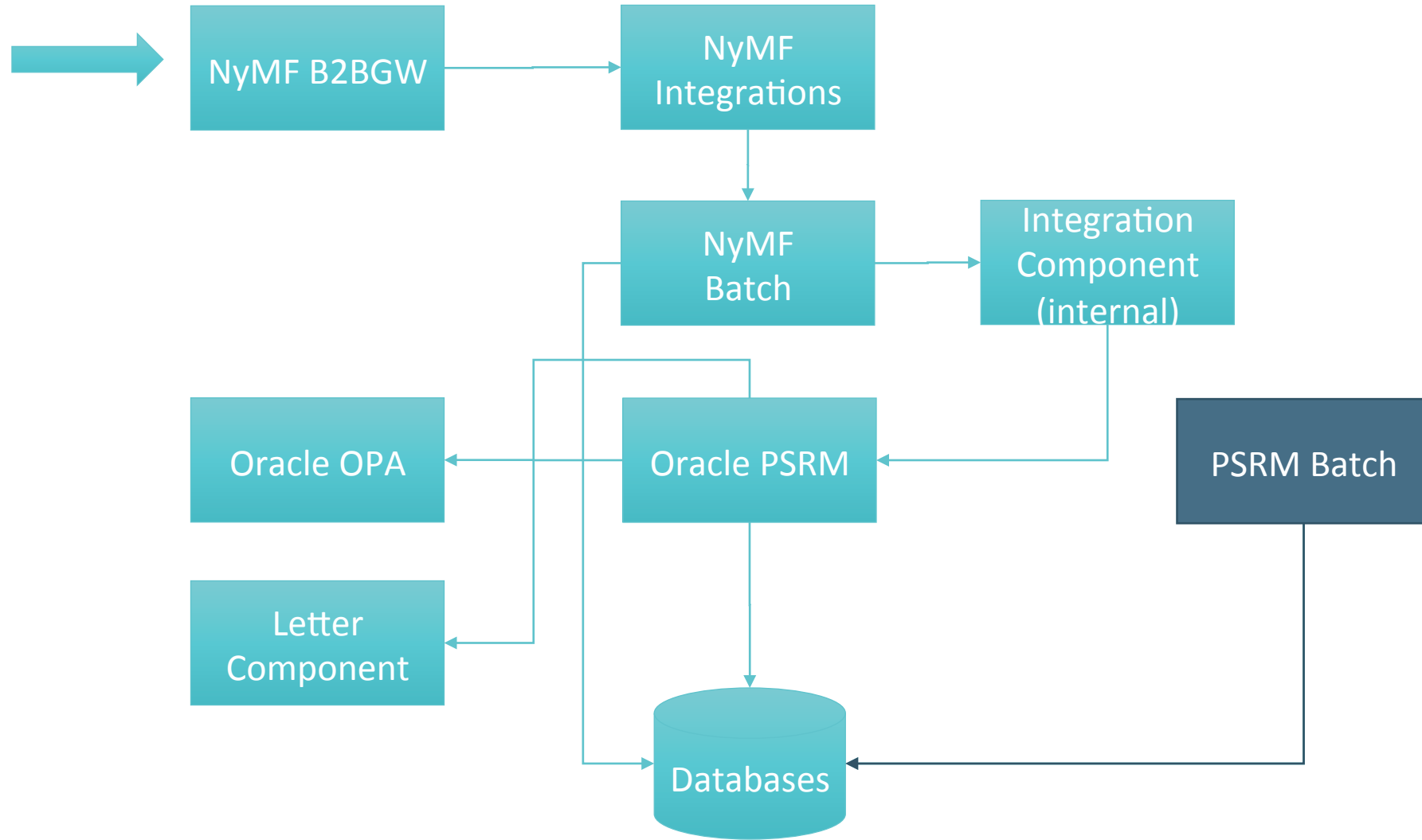
Lightning-course of ICI

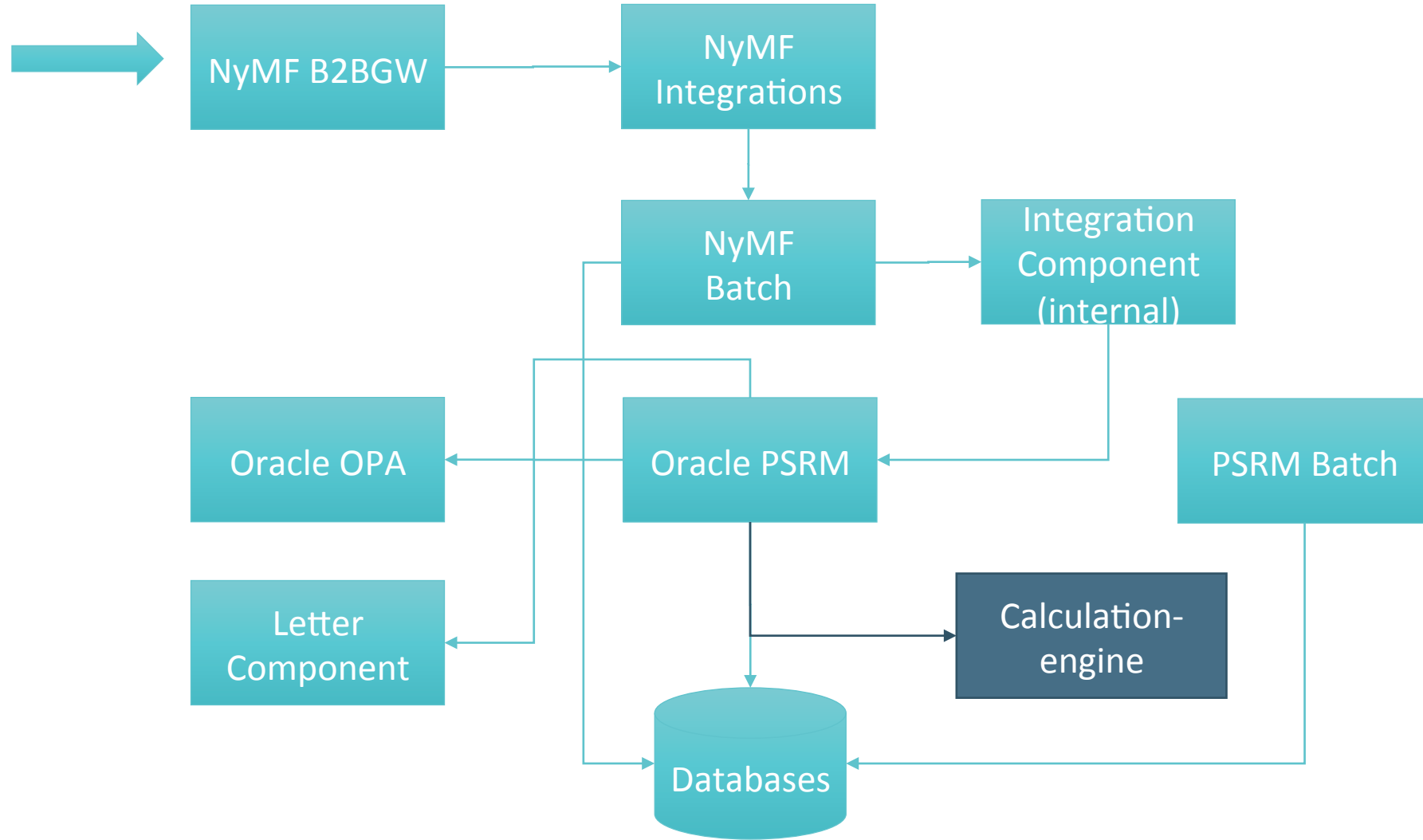


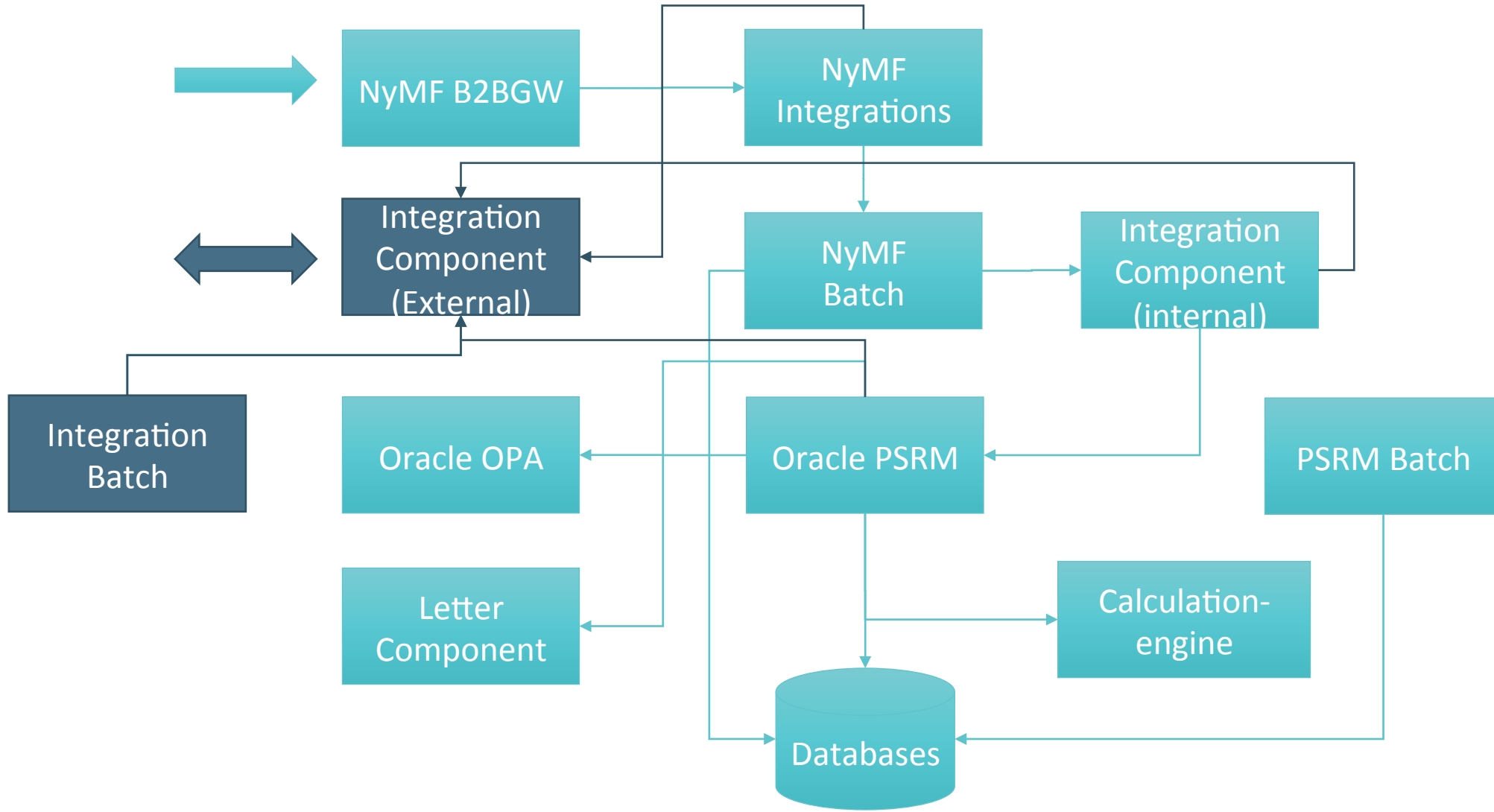


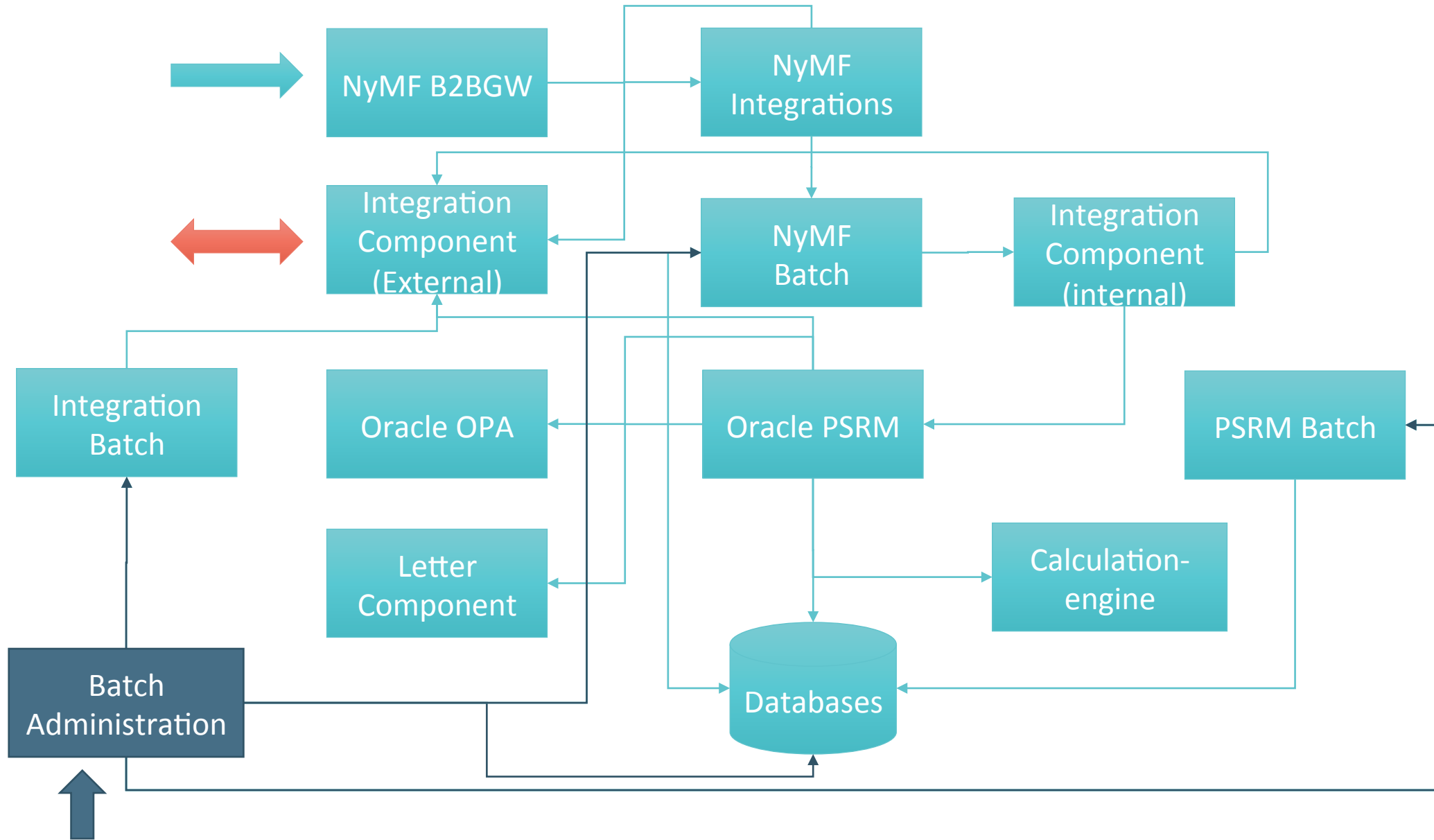


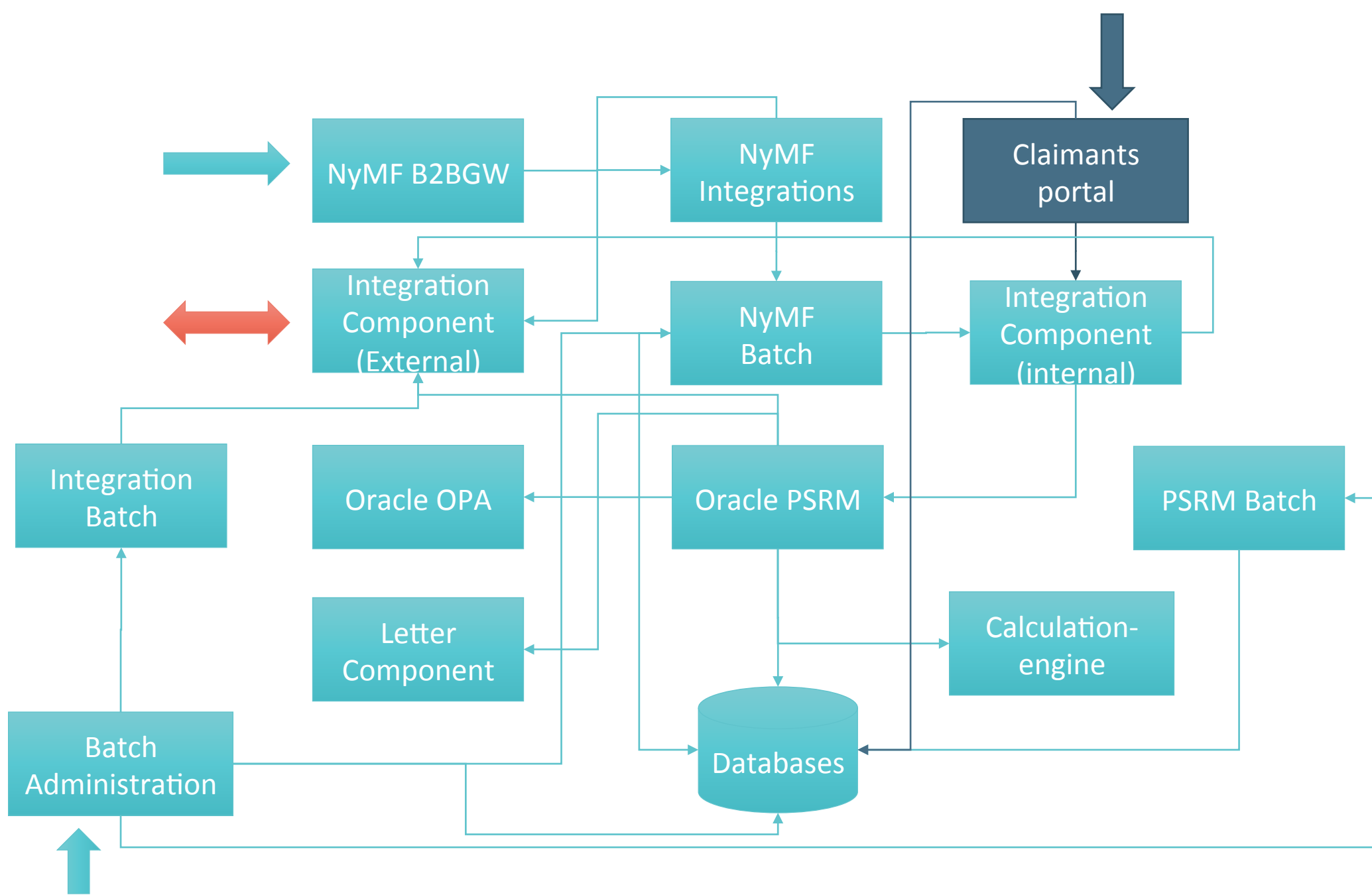


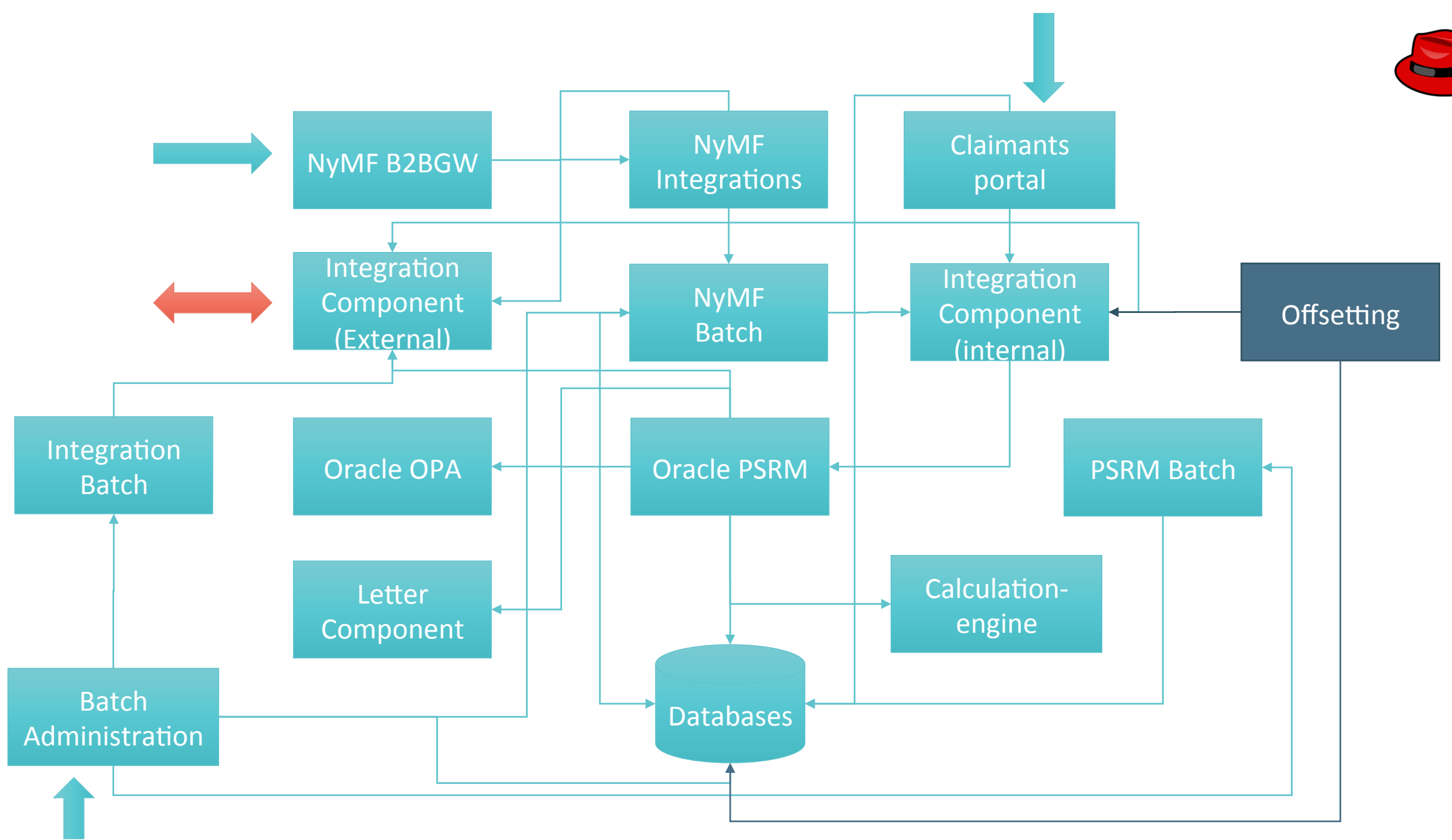


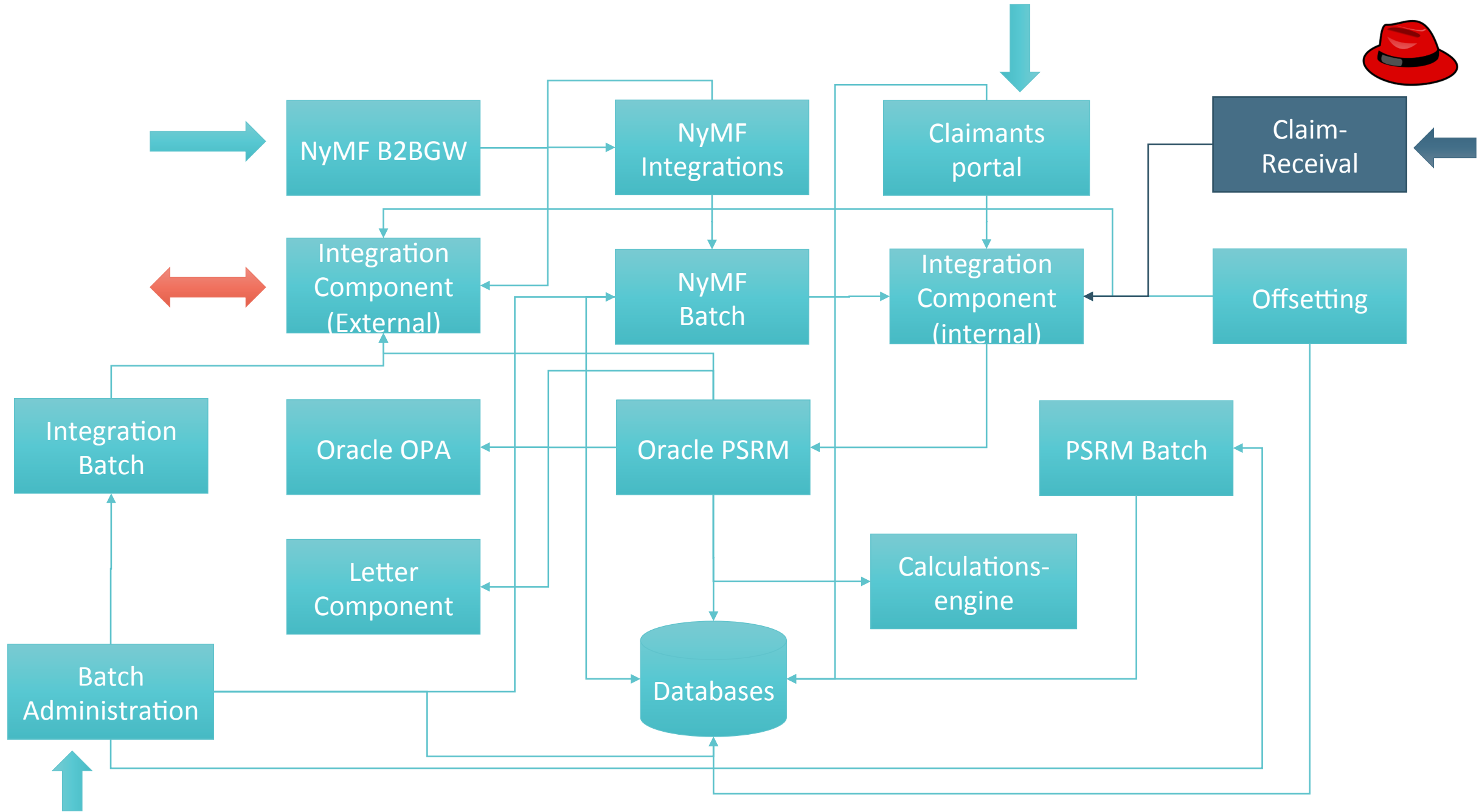








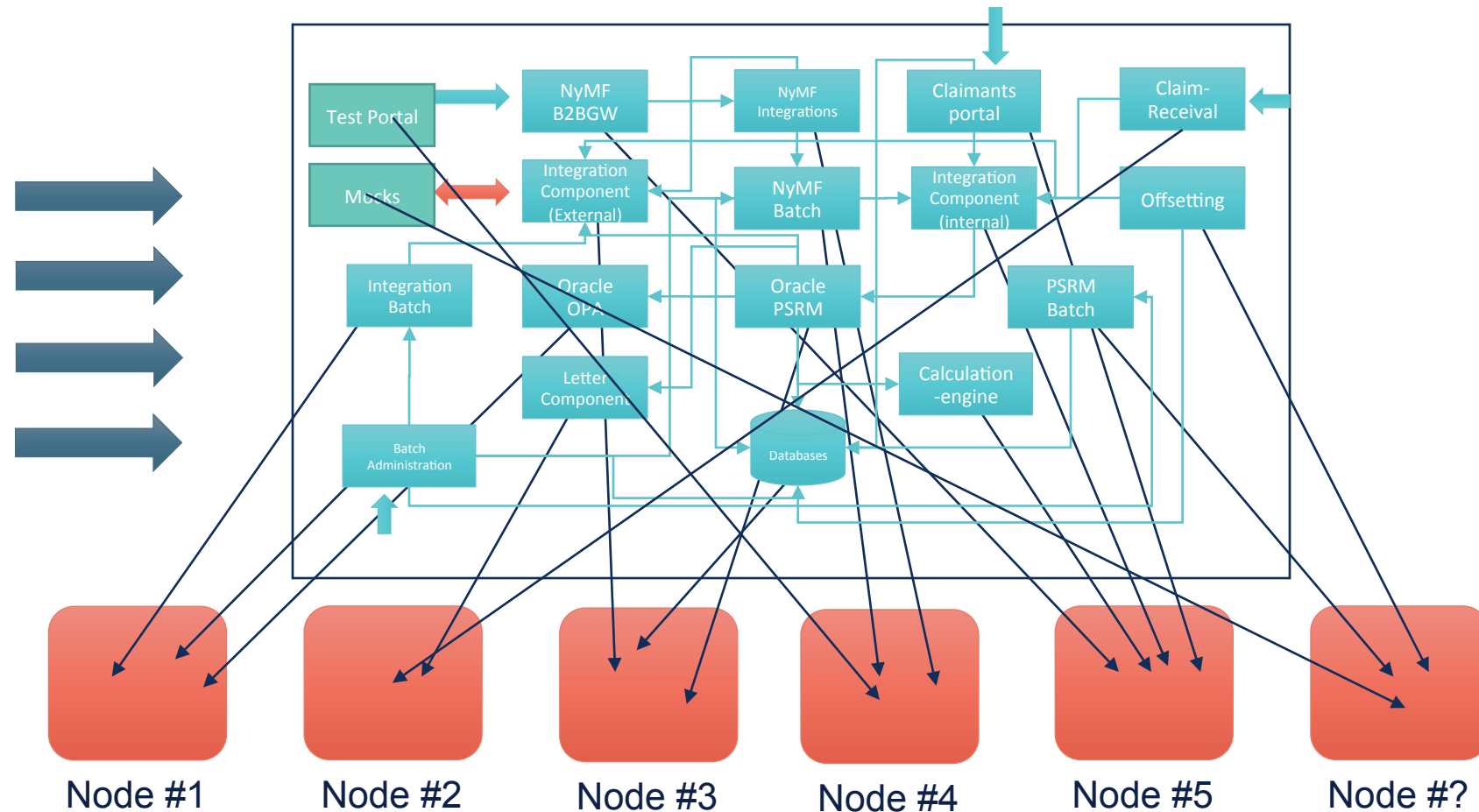






What happens as we provision the new environment?

1. Virtual network is configured
2. Containers is being transferred and started up on worker-nodes.
3. DNS is configured to all containers on preconfigured routes (practical if you want to reach an environment while you are developing locally).



Benefits and lessons learned using OpenShift

- **DIVISION:** We can divide our environments (projects) between teams, and setup user management between the projects (something that would be hard to do on Vanilla Kubernetes)
 - Can be important in especially a multi-vendor setup
- **ACCESIBILITY:** Easy to remember URL's for all applications, including endpoints that are normally not exposed to "public". We use a numeric name for each environment, ie. "dock01", "dock02". This way we can also expose db-ports on a deterministic port-number, making it easier for teammates to surf the environments without any support.
- **STABILITY:** Although there has been bumps on the road to put all components in containers, we see that our "docker-environments" has a lot more stability, and we see no unique strange behavior on them.
- **MEMORY:** Memory-management in OpenShift and Docker-containers have been difficult, and we are still tweaking to reduce our footprint.
 - We have reduced memory footprint from 65GB to 35GB per environment.
- **SCALEABILITY:** The setup scales nicely, although due to the very big memory-footprint, we need to be very proactive in ordering new hardware (the solution is hosted on-premise).
- It's been a learning proces, but it's been worth it and the program is saving hours in support already. It would, obviously, have been much better to do from the beginning of the project, but we didn't have this option.
- **It now takes ~22 minutes to provision a new environment with all applications running. This took days before hand. This enables us to have 60+ full environments running (yes, they are all being used).**

Final notes

- Being able to run automated quality assurance during the night and marking releases as stable for quick provisioning by teammembers, is crucial when you have a setup that relies a lot on manual test processing.
- This enables the teams themselves to order/reorder a fresh environment with the requirements they need.
 - Integrations/mocks
 - Enable or disable batch scheduling
- This orchestrated docker-setup increases stability and makes it possible for the technical teams in the project to focus on other tasks than environment-specific problems we experienced earlier.
- Accelerating this proces enables us to increase velocity in time2market and therefore begin harvesting debt at a faster pace, although of course this is not the only parameter in this area.

DEMO OF THE NEW ENVIRONMENT

Q&A

WE ARE COMMITTED



—
www.netcompany.com

Dan Strang Søndergaard
dss@netcompany.com
+45 61248564

netcompany

© Copyright 2018 Netcompany. All rights reserved.

Electronic, mechanical, photographic or other reproduction, translation or copying of this document or parts thereof, for whatever purpose, is not permitted without the prior written permission of Netcompany

APPENDIX

How is Netcompany different?

- IT people leading IT people
 - From CEO to project manager – all have strong IT background
 - To lead and manage complexity of modern, agile IT projects, you need to truly understand IT
- Relentless focus on delivery
 - Strong delivery culture & methodology
 - We empower our employees and expect them to take responsibility of quality and deadlines
- We are young
 - Average age is around 30
 - Talent, dedication & ambition to grow often trumps experience
- We celebrate successful projects & going into production – not sales!