RED HAT FORUM, SEPTEMBER 2018

DOCKER AND OPENSHIFT IN NY INDDRIVELSE (ICI)

Date: September 3, 2018
Version:
Author: Michael Vognsen Nielsen & Dan Strang Søndergaard
Contact: mvn@Netcompany.com & dss@netcompany.com

netcompany

Agenda

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

ABOUT US

Dan Strang Søndergaard

- Netcompany from 2006-2010 & 2014-now
- Microsoft: 2010-2014
- Has worked as developer, solution architect, project manager, pre-sales engineer & principal consultant
- Currently leading DevOps & cloud practice in Netcompany – including Red Hat
- Husband, father of two & runner

Michael Vognsen Nielsen

- Netcompany from 2015-now
- Has worked as developer, architect, pre-sale engineer, team lead
- Lead on Netcompany DevOps Center of Excellence
- Father of two

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.

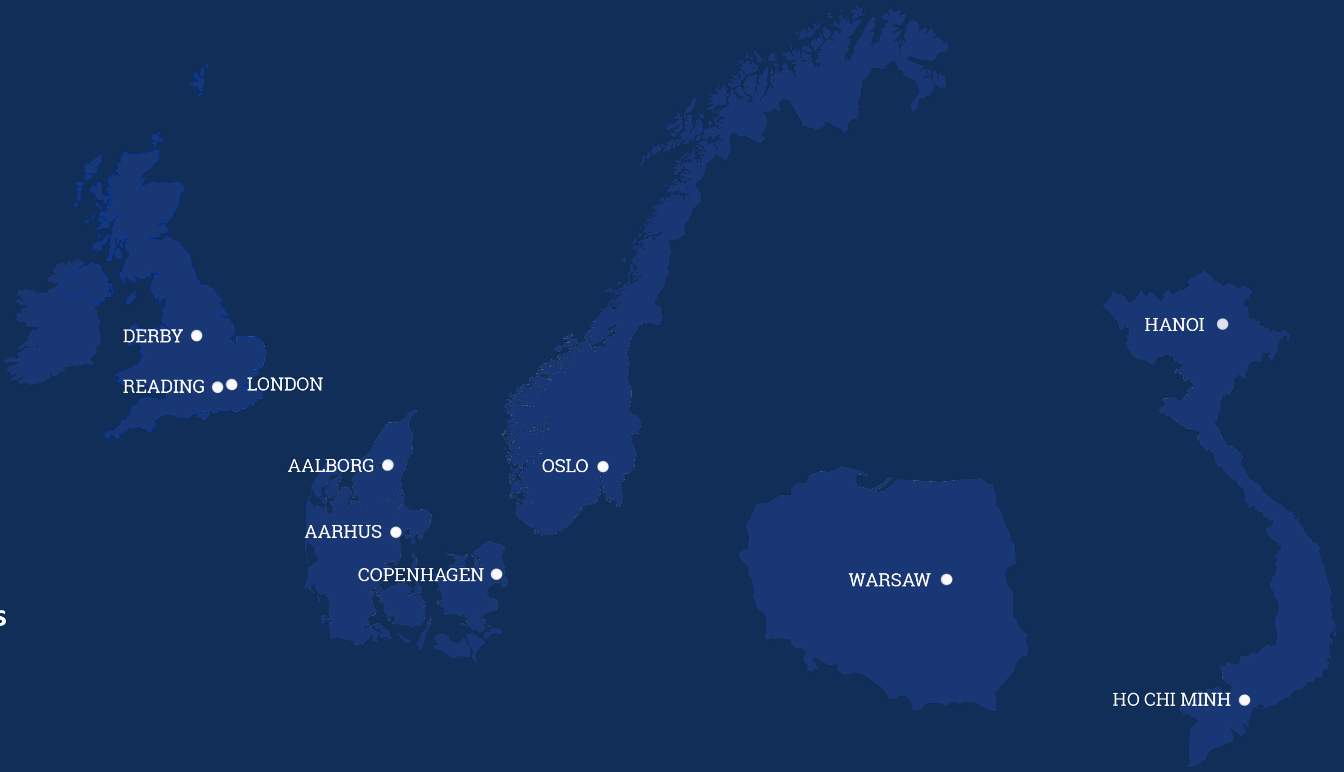
Netcompany was founded in 2000.

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

Microsoft .NET

redhat

amazon.com

Azure

Throughout your digital journey…

Innovation

Development

Operation

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
  - We believe talent, dedication & willingness to grow trump experience

- We celebrate successful projects & going into production – not sales

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 legislation, so two years later the system was shut-down permanently
- At that point the amount of unpaid debt had risen to 100 BILLION DKK
  - Additionally, SKAT had been subject to fraud that cost SKAT more than 12 BILLION DKK due to poor oversight of refund for tax on yields to foreign investors
- 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

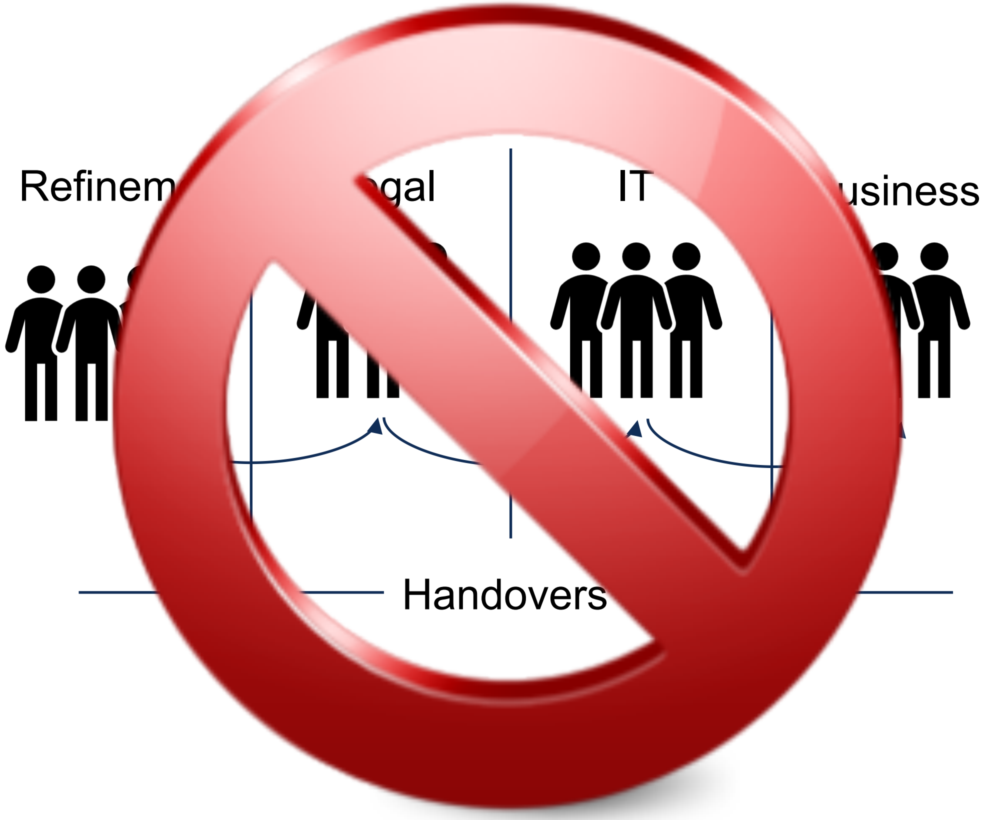
- The first release went live in May 2017, and Denmark could start collecting debt again
  - We are not done
  - More public authorities are being onboarded to accelerate debt collection

- Netcompany has 150 full time consultants working on the project & with SKAT and other vendors the project is 300 people

- Netcompany uses SAFe™ to manage and govern the project
  - 15 agile, cross functional teams

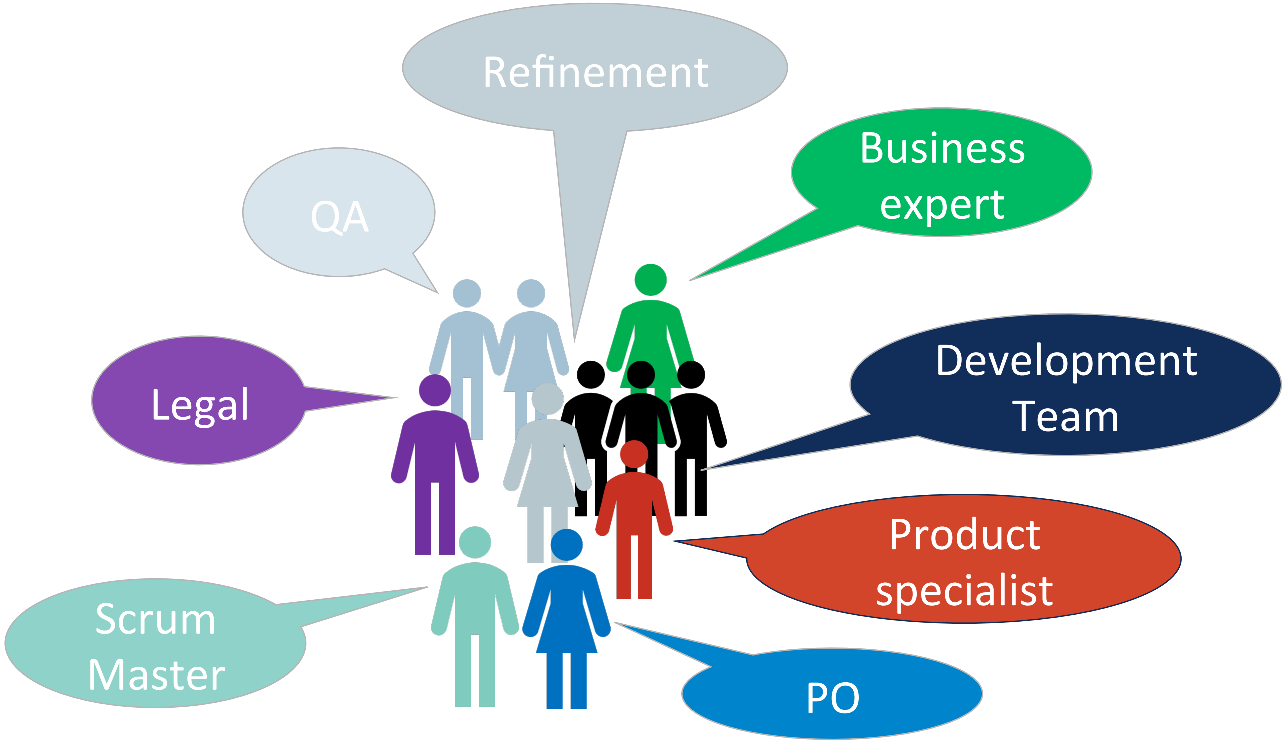
- All of this makes this project particular demanding in terms of the underlying IT platform

ICI | Agile teams



- No ownership
- Immense communicative overhead
- Difficult planning
- Blame games
- Stories being sent back and forth

ICI | Agile teams



Roles
SKAT, 1 person per team
SKAT, 1 til 2 per team
Accenture, 2 per team
Netcompany, 1 per team
Kammeradvokaten, 1 til 2 per team
Netcompany, Deloitte, Accenture, SKAT – 3 per team
Netcompany, 4 per team
Netcompany, 1 shared among all teams

DEMO – PROVISIONING A NEW ENVIRONMENT

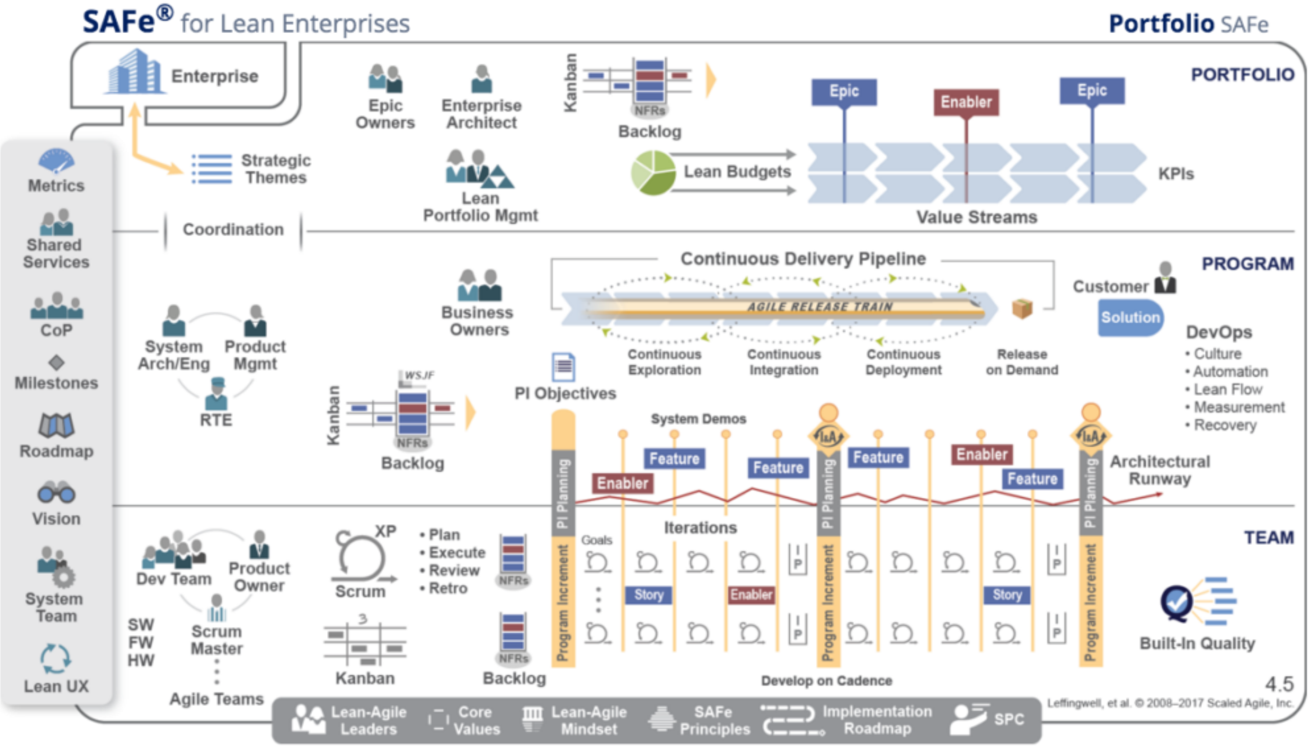
Introduction of the ICI project

- Agile projekt based on Scaled Agile Framework (SAFe). I was teamlead for two technical enabler-teams.
- Replacement of ”old” EFI/DMI system that were deprecated due to a large number of problems.
    - Automated debt-recovery was turned off in the fall of 2015.
- 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).

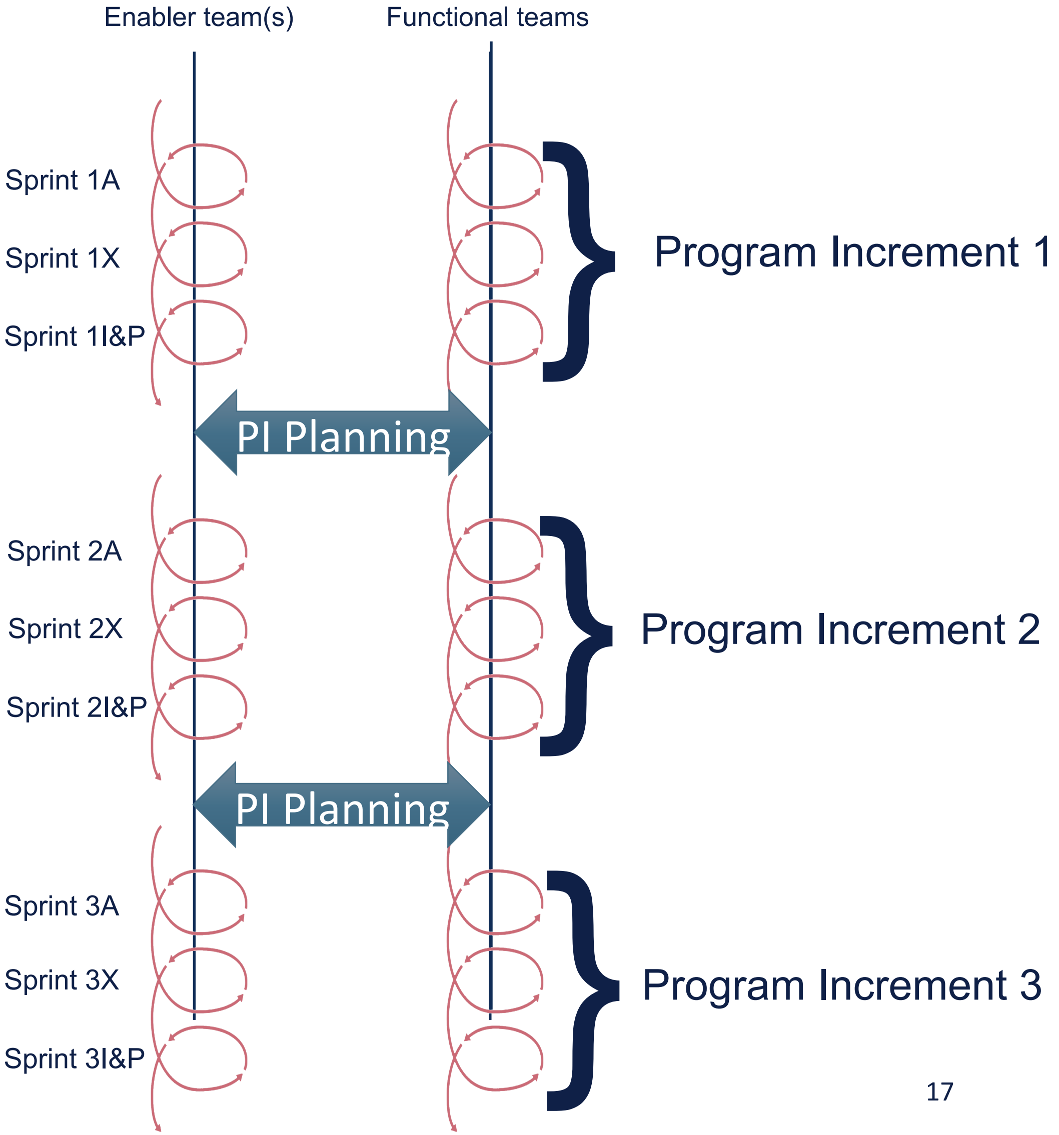
“Maybe in a sense we're reducing debt.”

President Donald Trump

What is SAFe?



- Maintain hard syncronization between teams at PI Planning
  - This is where we, in rough terms, commit us to work the next Program Increment (ie. 5 sprints)
- Focus on Continuous Delivery to be able to Release on Demand.



Enabler team(s)

Functional teams

Sprint 1A

Sprint 1X

Sprint 1I&P

Program Increment 1

PI Planning

Sprint 2A

Sprint 2X

Sprint 2I&P

Program Increment 2

PI Planning

Sprint 3A

Sprint 3X

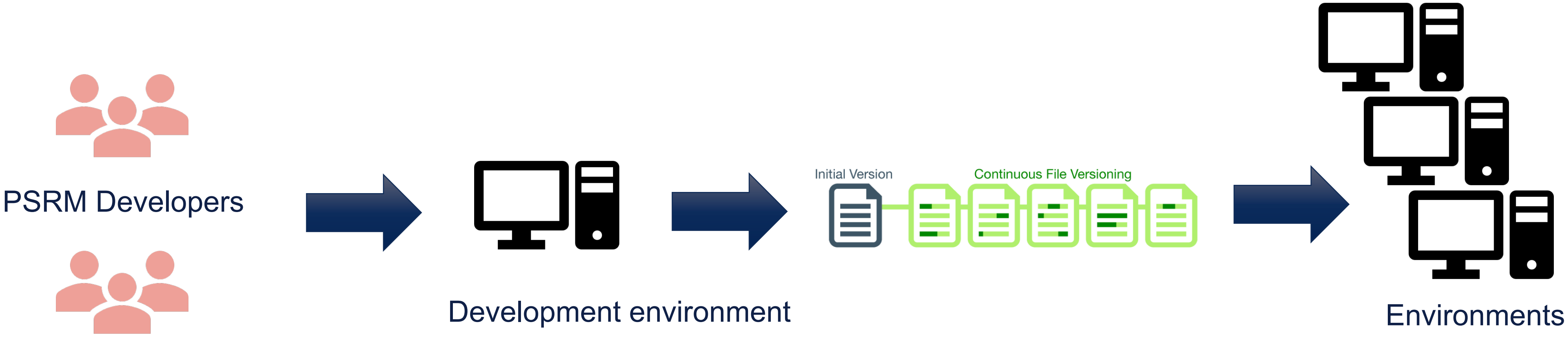
Sprint 3I&P

Program Increment 3

CONFIGURATION MANAGEMENT AND DEVELOPMENT OF PSRM

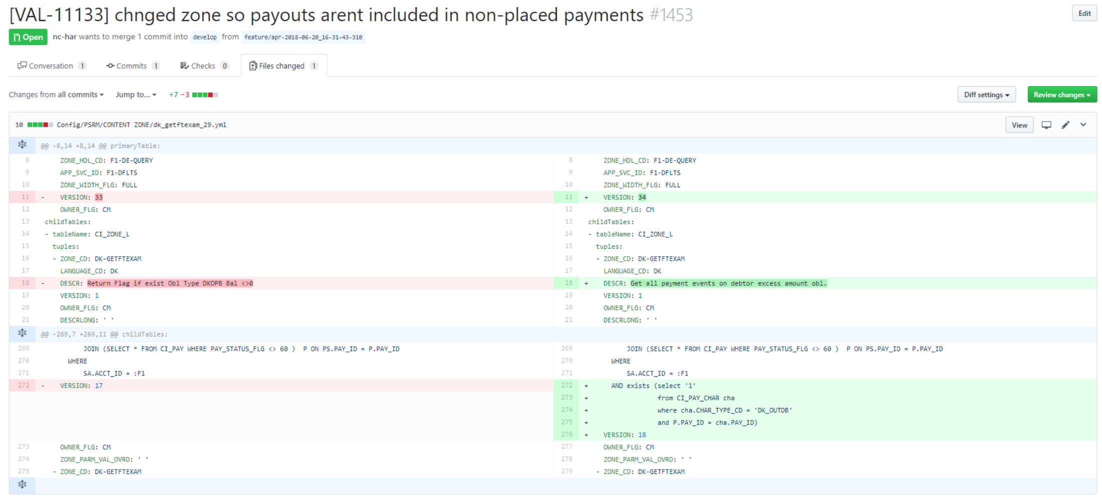
Configuration Management on ICI - Intro

- 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 (Maintainance 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] chnged zone so payouts arent included in non-placed payments #1453

nc-har wants to merge 1 commit into develop from feature/apr-2018-06-20_16-31-43-310

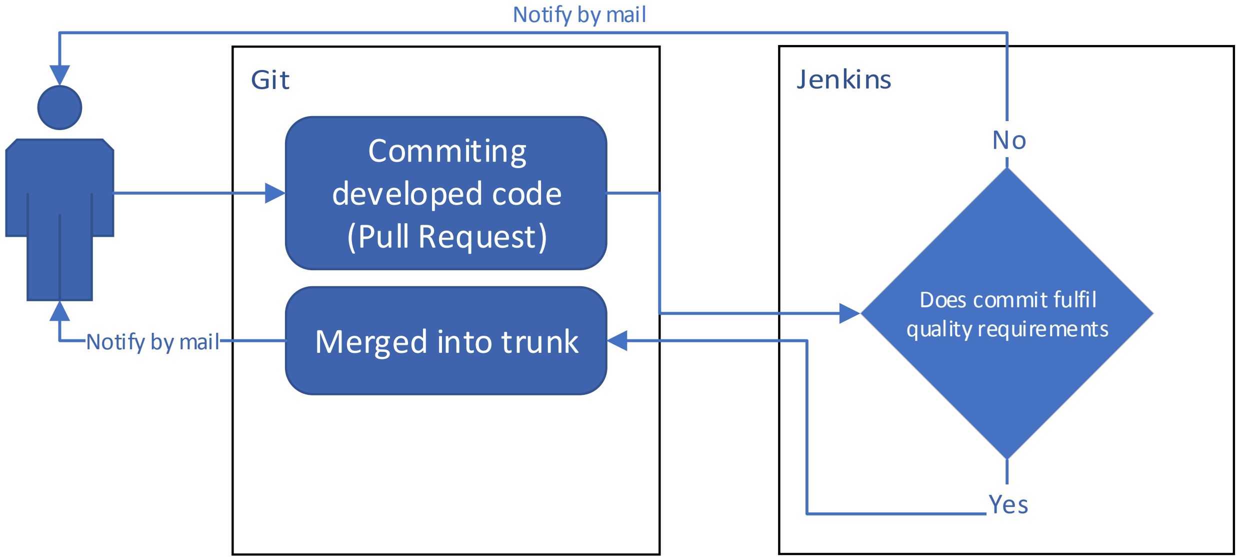
+7 −3

10 Config/PSRM/CONTENT ZONE/dk_getftexam_29.yml

```
@@ -8,14 +8,14 @@ primaryTable:
      ZONE_HDL_CD: F1-DE-QUERY
      APP_SVC_ID: F1-DFLTS
      ZONE_WIDTH_FLG: FULL
-     VERSION: 33
+     VERSION: 34
      OWNER_FLG: CM
  childTables:
  - tableName: CI_ZONE_L
    tuples:
    - ZONE_CD: DK-GETFTEXAM
      LANGUAGE_CD: DK
-     DESCR: Return Flag if exist Obl Type DKOPB Bal <>0
+     DESCR: Get all payment events on debtor excess amount obl.
      VERSION: 1
      OWNER_FLG: CM
      DESCRLONG: ' '
@@ -269,7 +269,11 @@ childTables:
          JOIN (SELECT * FROM CI_PAY WHERE PAY_STATUS_FLG <> 60 )  P ON PS.PAY_ID = P.PAY_ID
        WHERE
          SA.ACCT_ID = :F1
-     VERSION: 17
+       AND exists (select '1'
+                 from CI_PAY_CHAR cha
+                 where cha.CHAR_TYPE_CD = 'DK_OUTDB'
+                 and P.PAY_ID = cha.PAY_ID)
+     VERSION: 18
      OWNER_FLG: CM
      ZONE_PARM_VAL_OVRD: ' '
    - ZONE_CD: DK-GETFTEXAM
```

Automatiske quality gates: Pull Requests på 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



Notify by mail

Git

Commiting developed code (Pull Request)

Merged into trunk

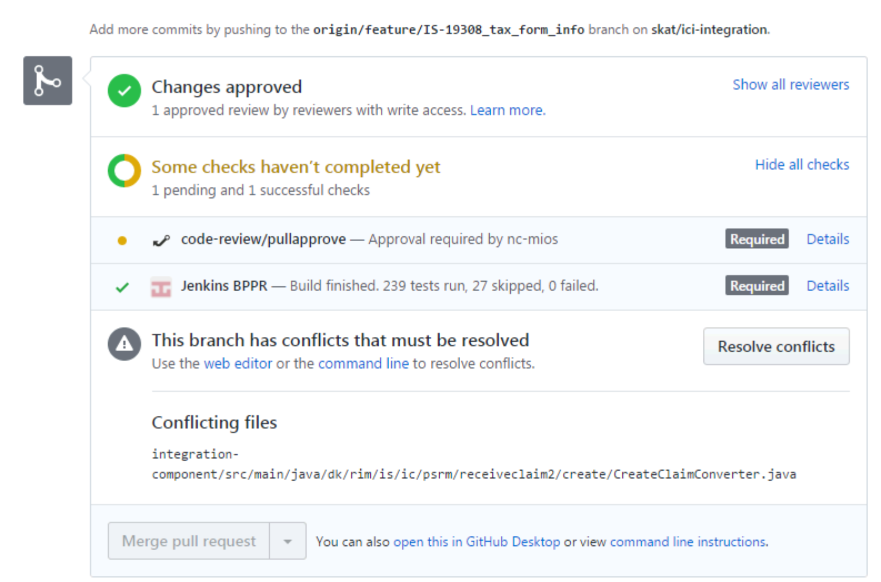
Notify by mail

Jenkins

No

Does commit fulfil quality requirements

Yes



Add more commits by pushing to the **origin/feature/IS-19308_tax_form_info** branch on **skat/ici-integration**.

✓ Changes approved — Show all reviewers

1 approved review by reviewers with write access. Learn more.

Some checks haven't completed yet — Hide all checks

1 pending and 1 successful checks

● code-review/pullapprove — Approval required by nc-mios — Required — Details

✓ Jenkins BPPR — Build finished. 239 tests run, 27 skipped, 0 failed. — Required — Details

This branch has conflicts that must be resolved — Resolve conflicts

Use the web editor or the command line to resolve conflicts.

Conflicting files

```
integration-
component/src/main/java/dk/rim/is/ic/psrm/receiveclaim2/create/CreateClaimConverter.java
```

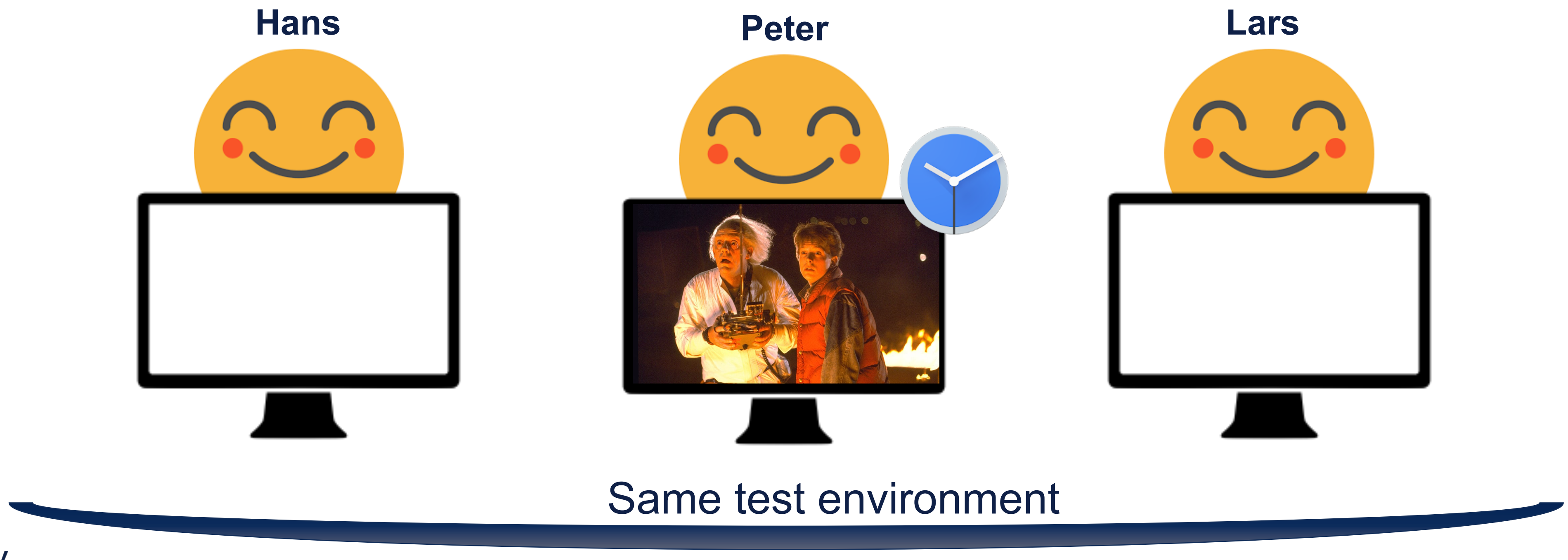
Merge pull request ▾ — You can also open this in GitHub Desktop or view command line instructions.

WHY DOCKER IN ICI?

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

Why Docker in ICI?

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



Hans

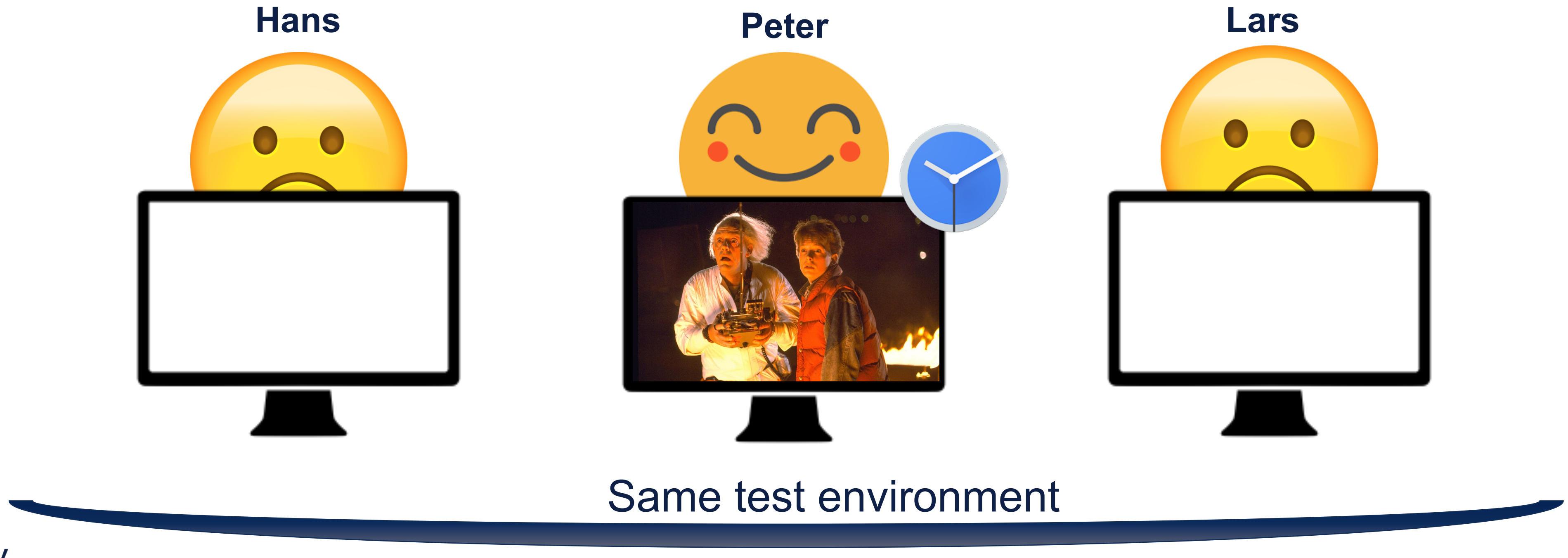
Peter

Lars

Same test environment

Why Docker in ICI?

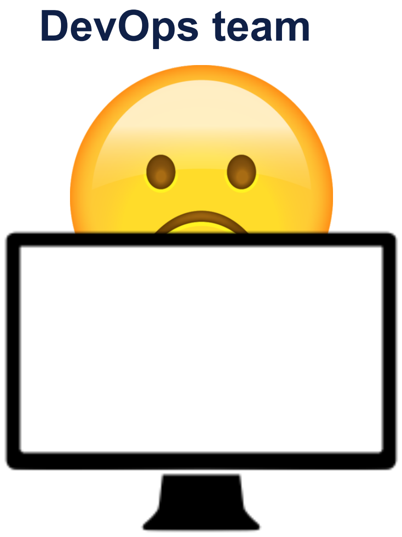
- **Test**
  - Verification (ordinary. Test)
  - Validation (Making sure that the system is legal)



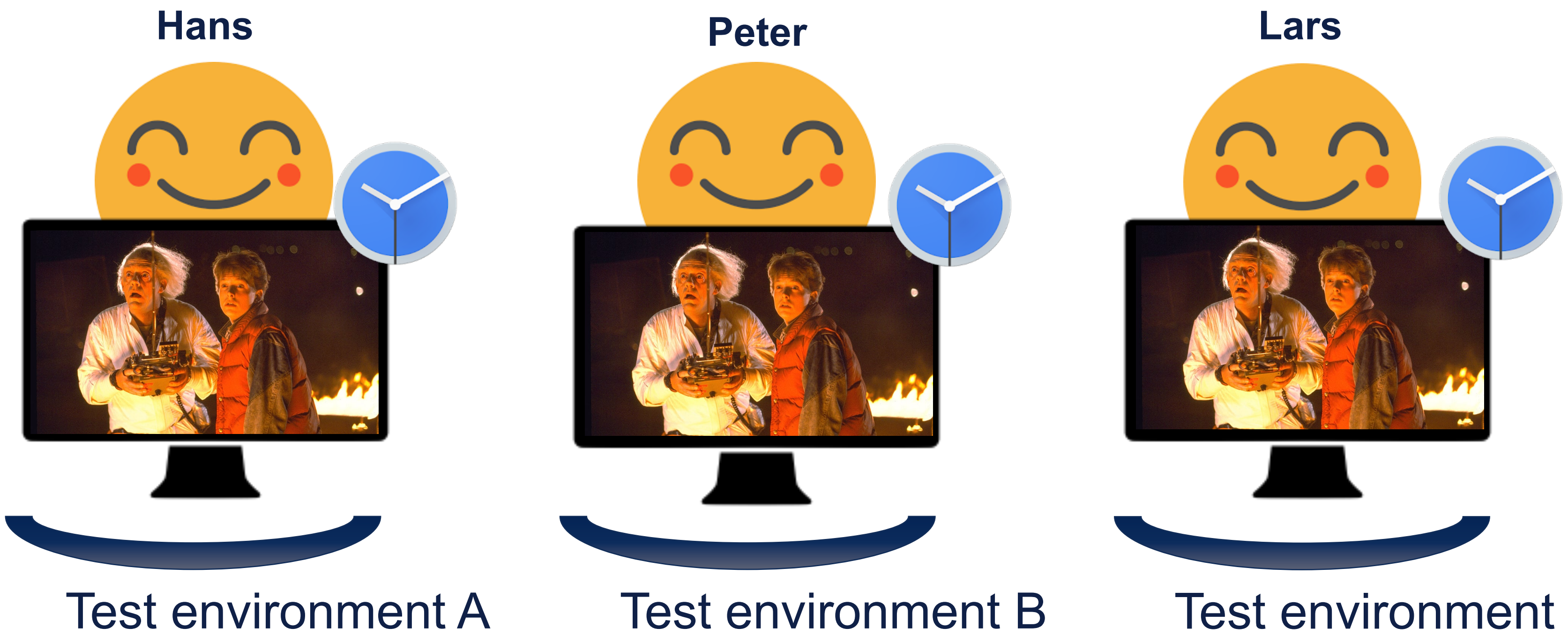
Same test environment

Why Docker in ICI?

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



DevOps team



Hans

Peter

Lars

Test environment A

Test environment B

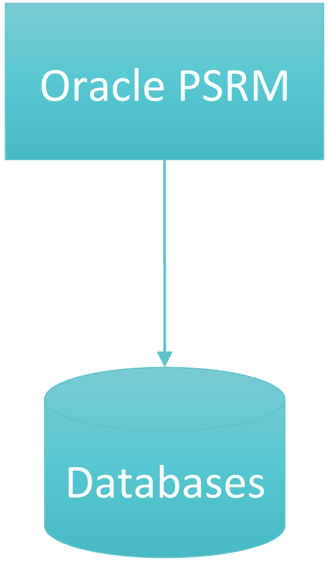
Test environment C

Why Docker 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
  - <u>We committed us to Docker in november 2017</u>

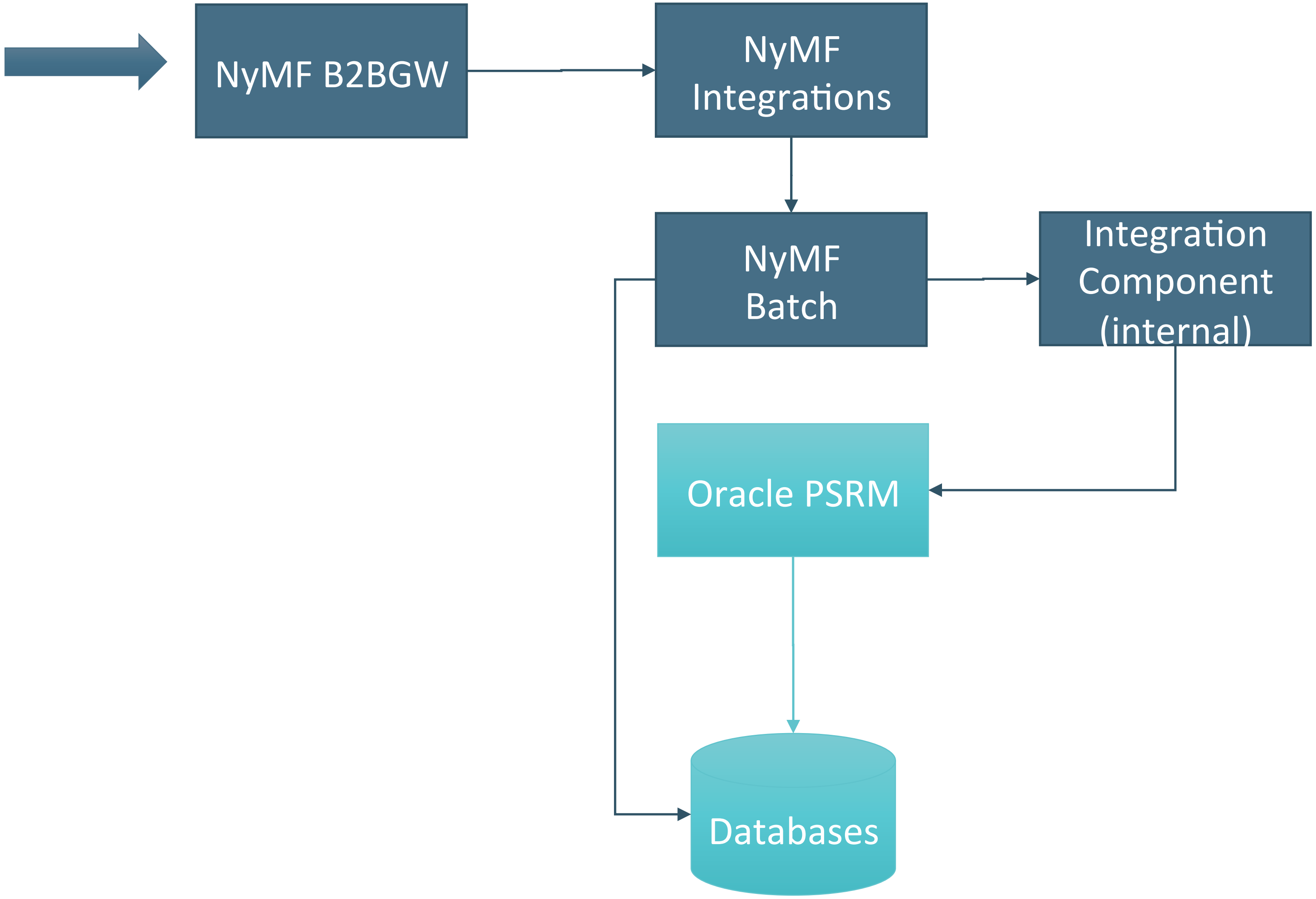
OPENSHIFT AND DOCKER IN ICI – WHAT HAPPENS WHEN WE PROVISION A NEW ENVIRONMENT?

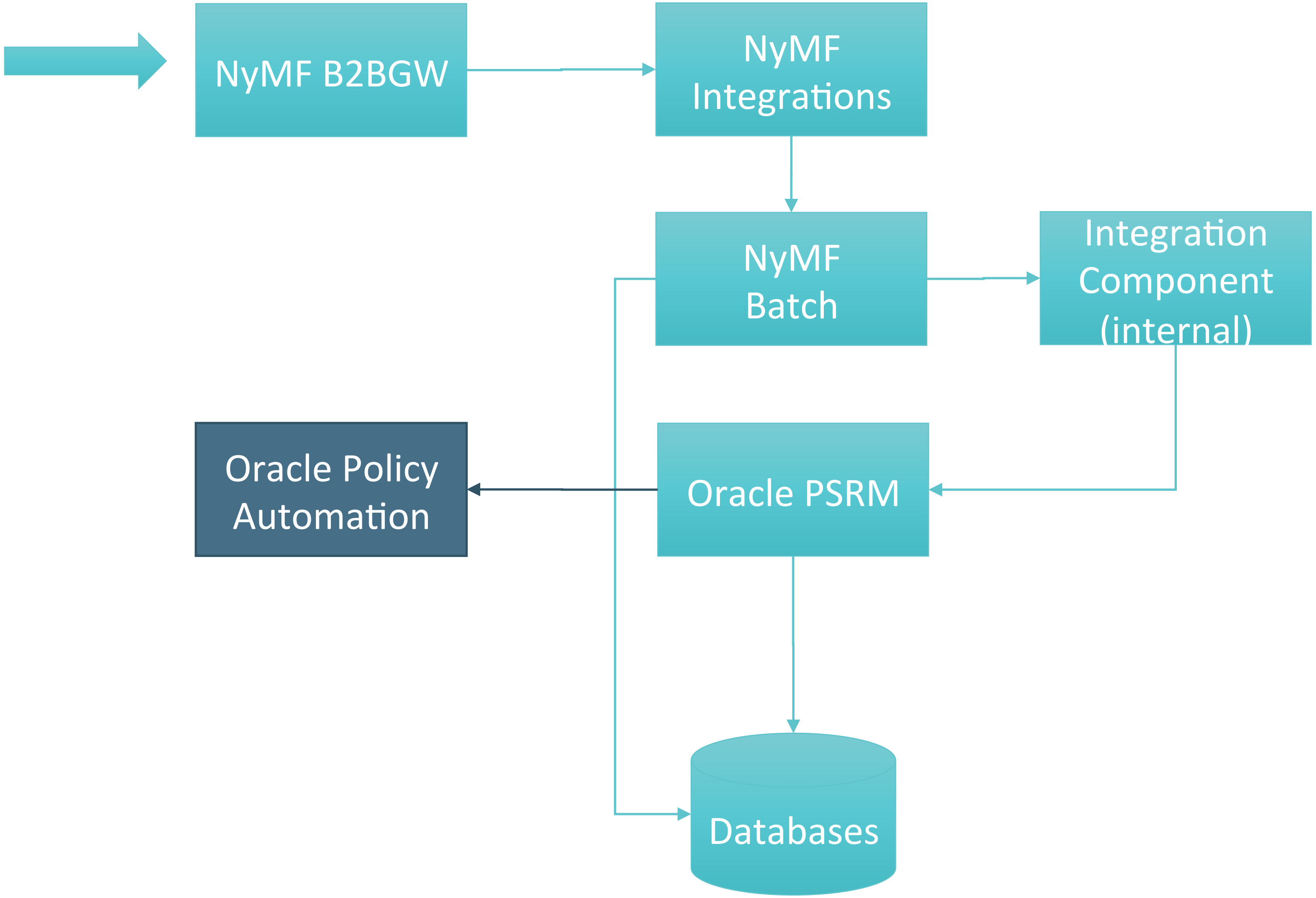
Lightning-course of ICI

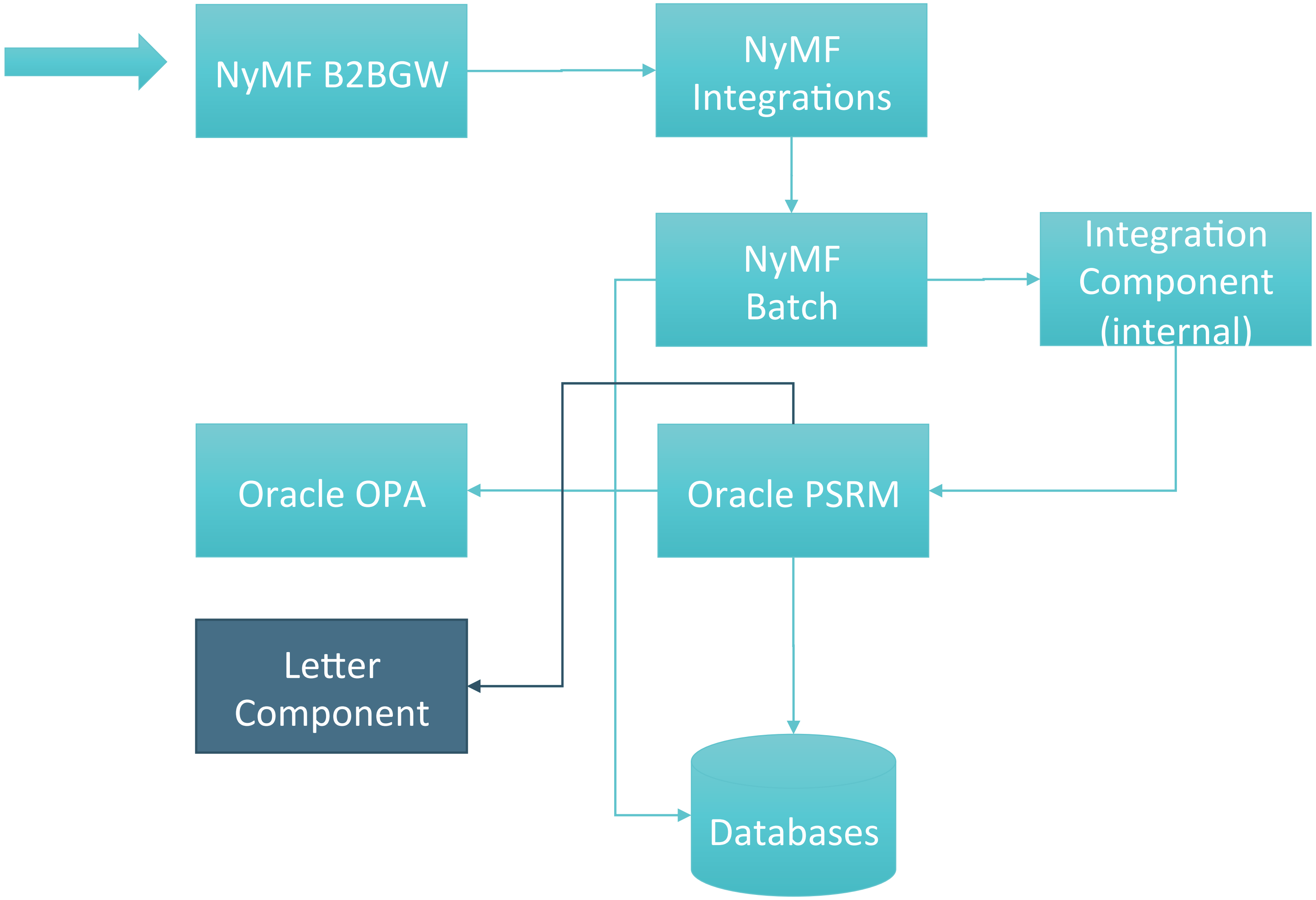


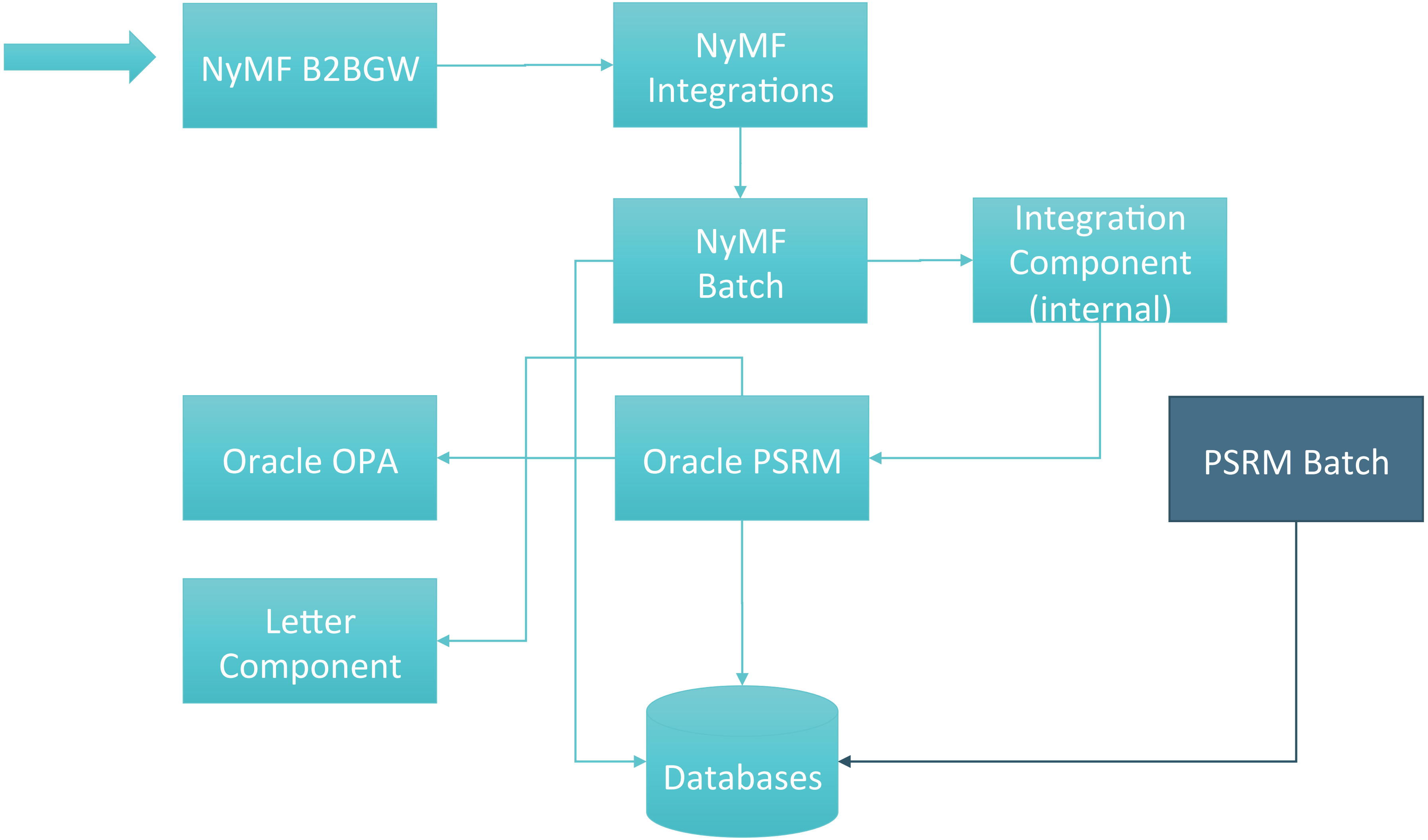
Oracle PSRM

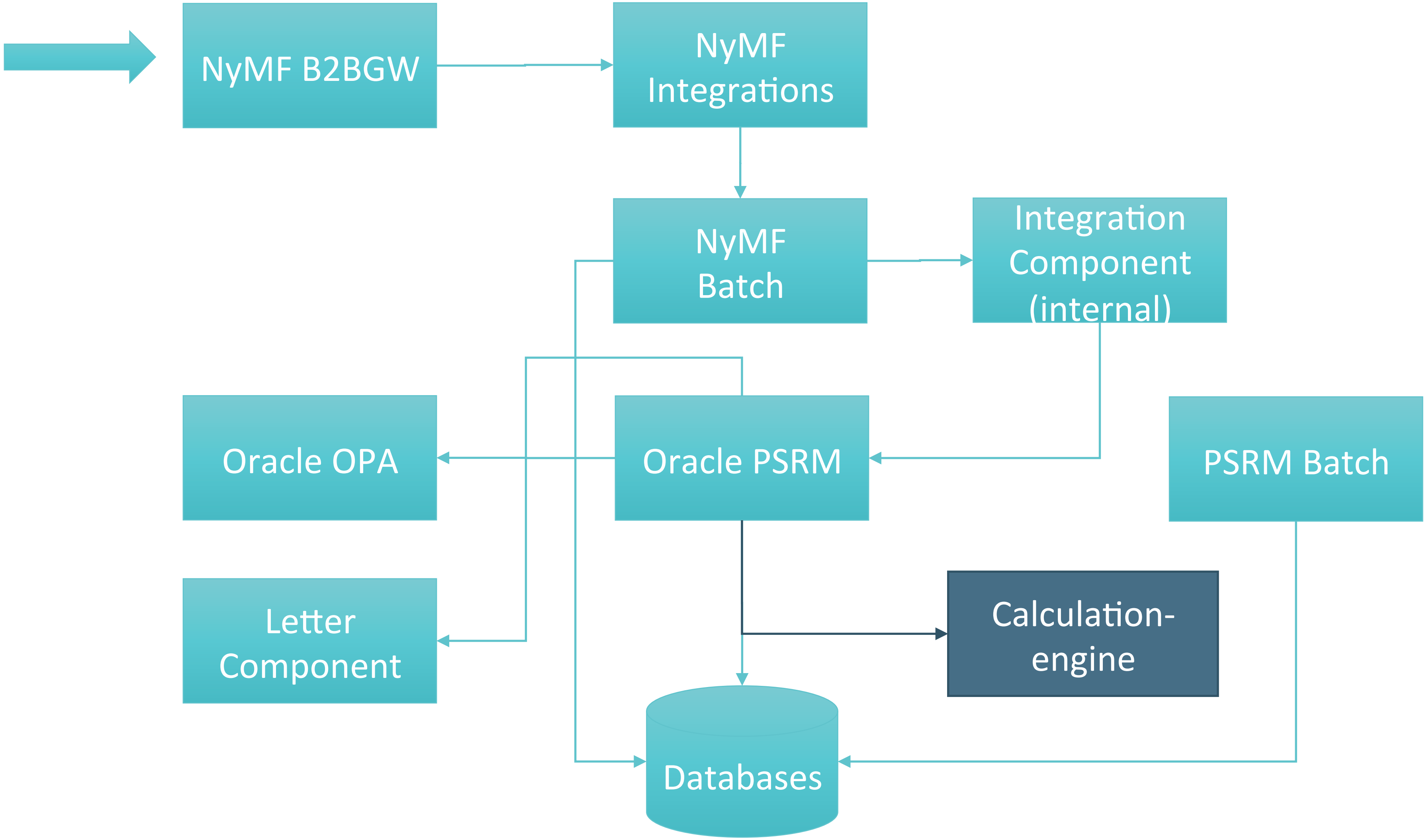
Databases

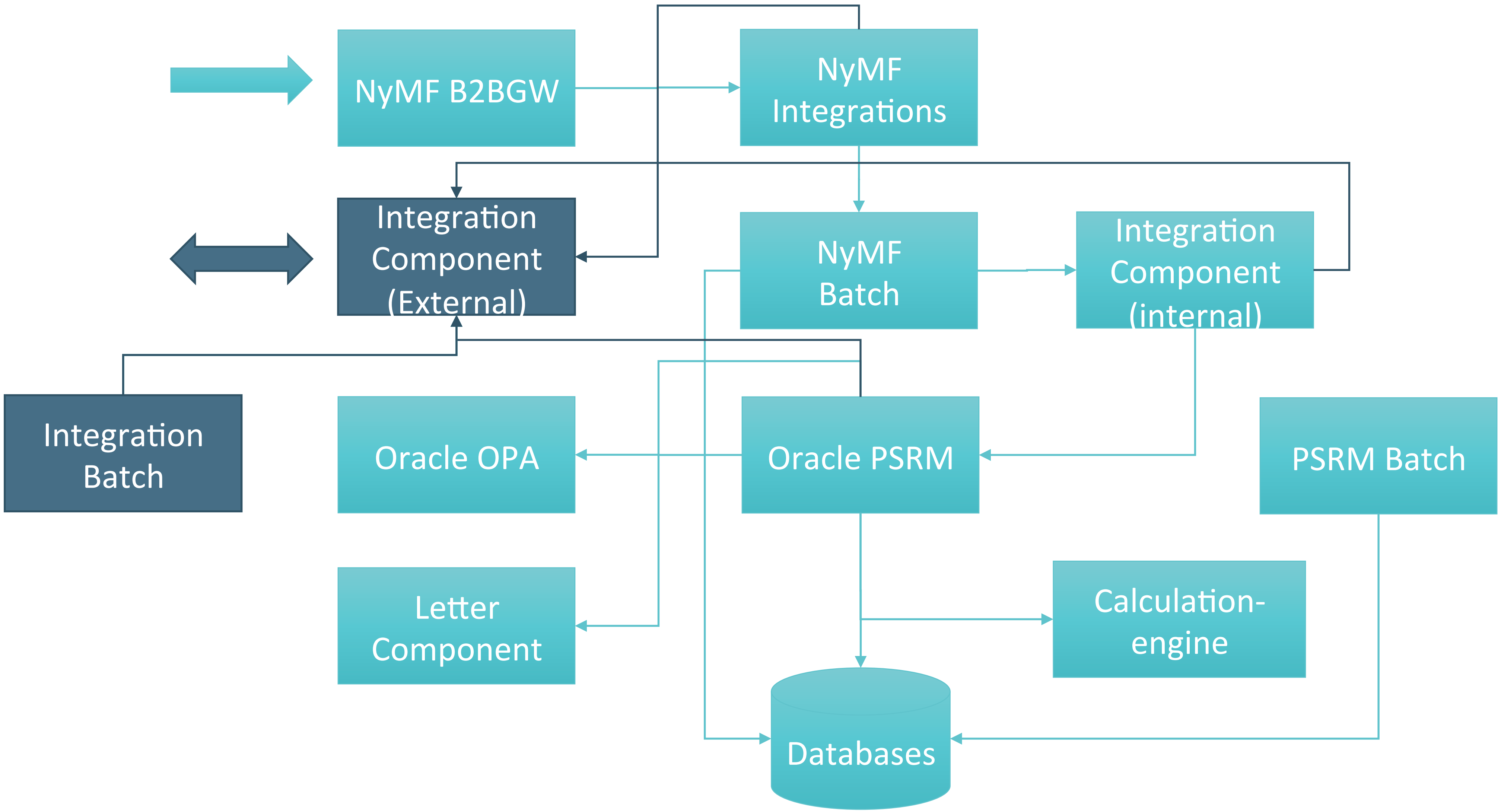

NyMF B2BGW
NyMF Integrations
NyMF Batch
Integration Component (internal)
Oracle PSRM
Databases

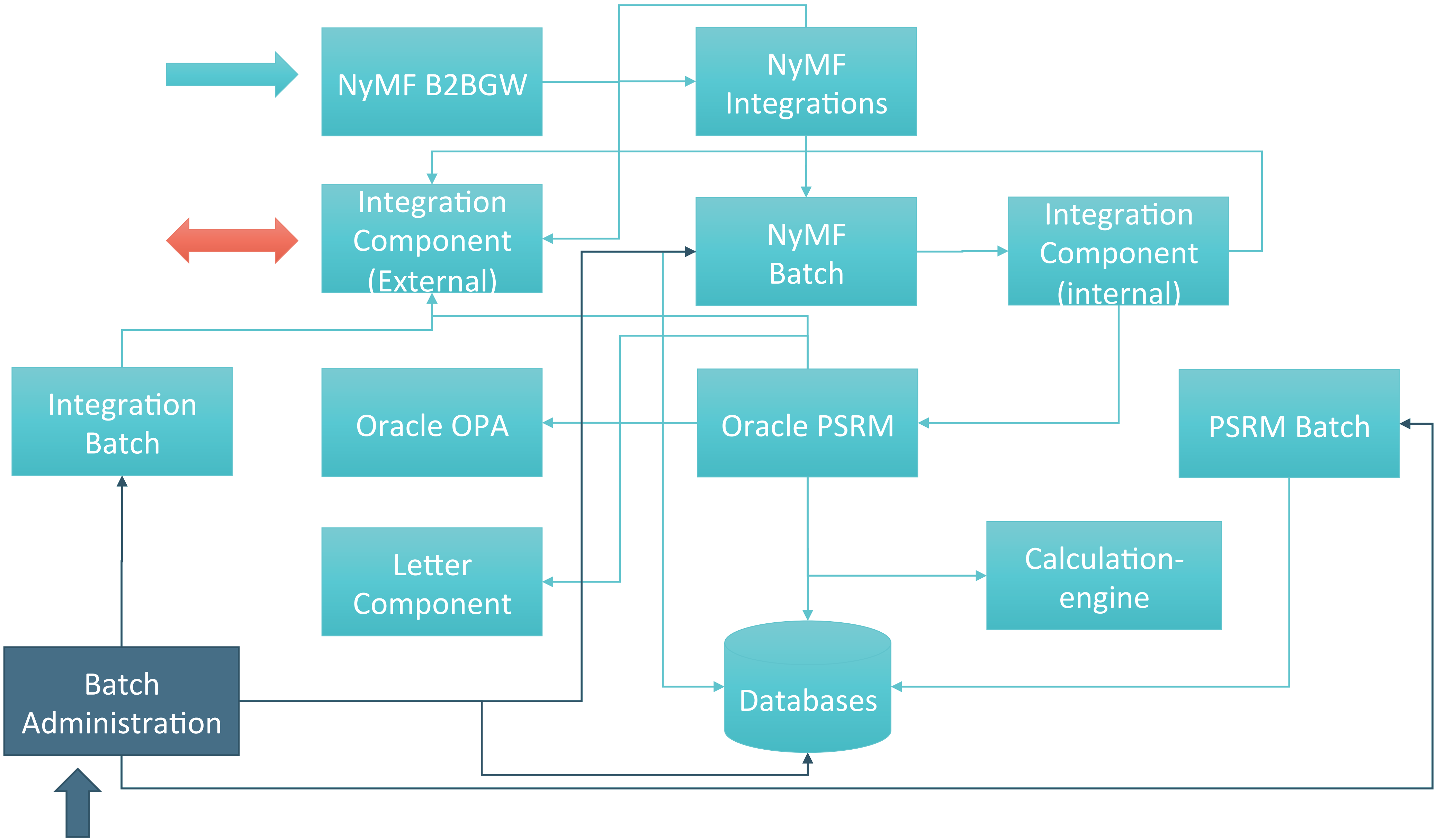

NyMF B2BGW
NyMF Integrations
NyMF Batch
Integration Component (internal)
Oracle Policy Automation
Oracle PSRM
Databases

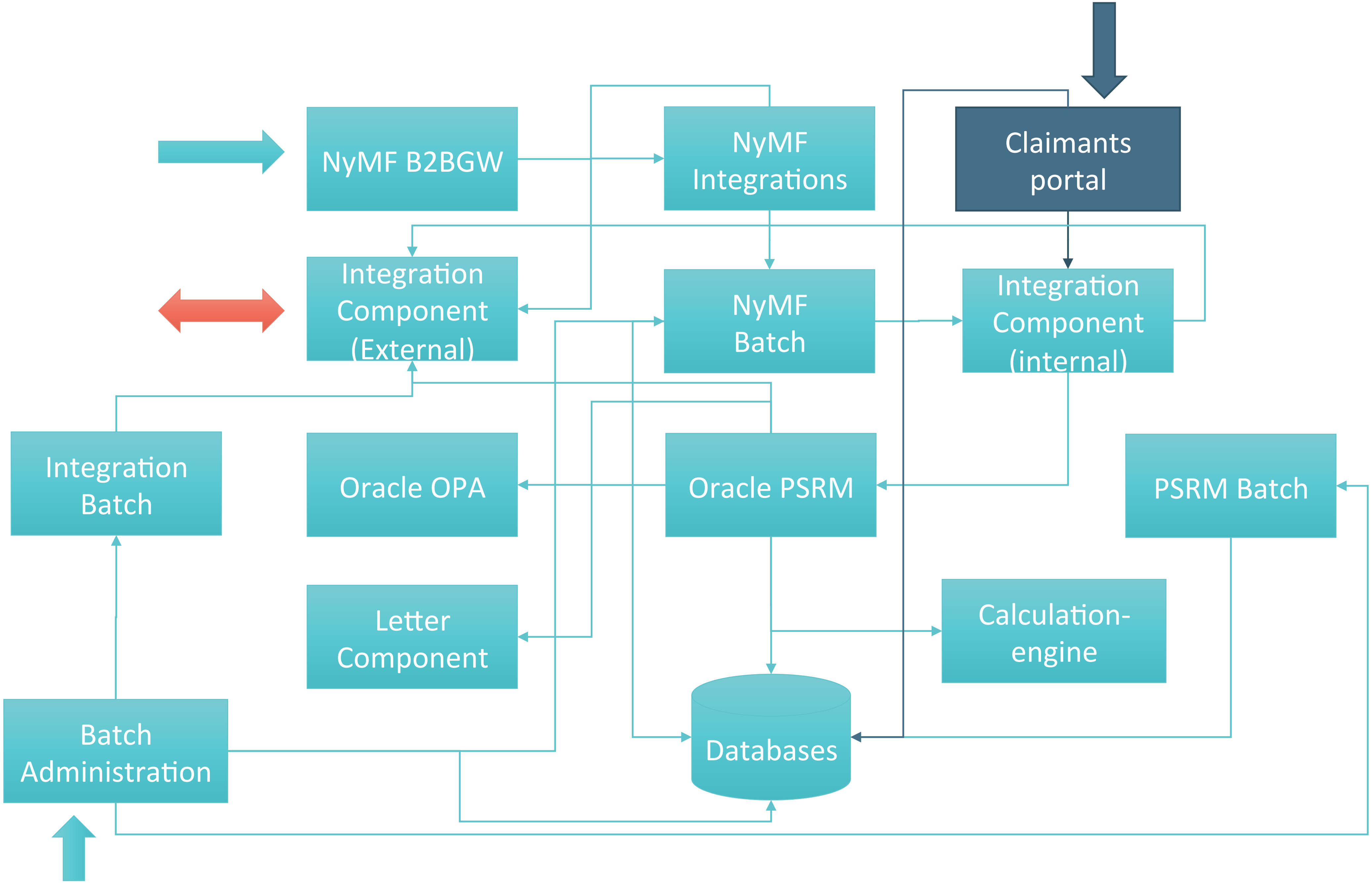

NyMF B2BGW
NyMF Integrations
NyMF Batch
Integration Component (internal)
Oracle OPA
Oracle PSRM
Letter Component
Databases


NyMF B2BGW
NyMF Integrations
NyMF Batch
Integration Component (internal)
Oracle OPA
Oracle PSRM
PSRM Batch
Letter Component
Databases


NyMF B2BGW
NyMF Integrations
NyMF Batch
Integration Component (internal)
Oracle OPA
Oracle PSRM
PSRM Batch
Letter Component
Calculation-engine
Databases


NyMF B2BGW
NyMF Integrations
Integration Component (External)
NyMF Batch
Integration Component (internal)
Integration Batch
Oracle OPA
Oracle PSRM
PSRM Batch
Letter Component
Calculation-engine
Databases


NyMF B2BGW
NyMF Integrations
Integration Component (External)
NyMF Batch
Integration Component (internal)
Integration Batch
Oracle OPA
Oracle PSRM
PSRM Batch
Letter Component
Calculation-engine
Databases
Batch Administration



NyMF B2BGW

NyMF Integrations

Claimants portal

Integration Component (External)

NyMF Batch

Integration Component (internal)

Integration Batch

Oracle OPA

Oracle PSRM

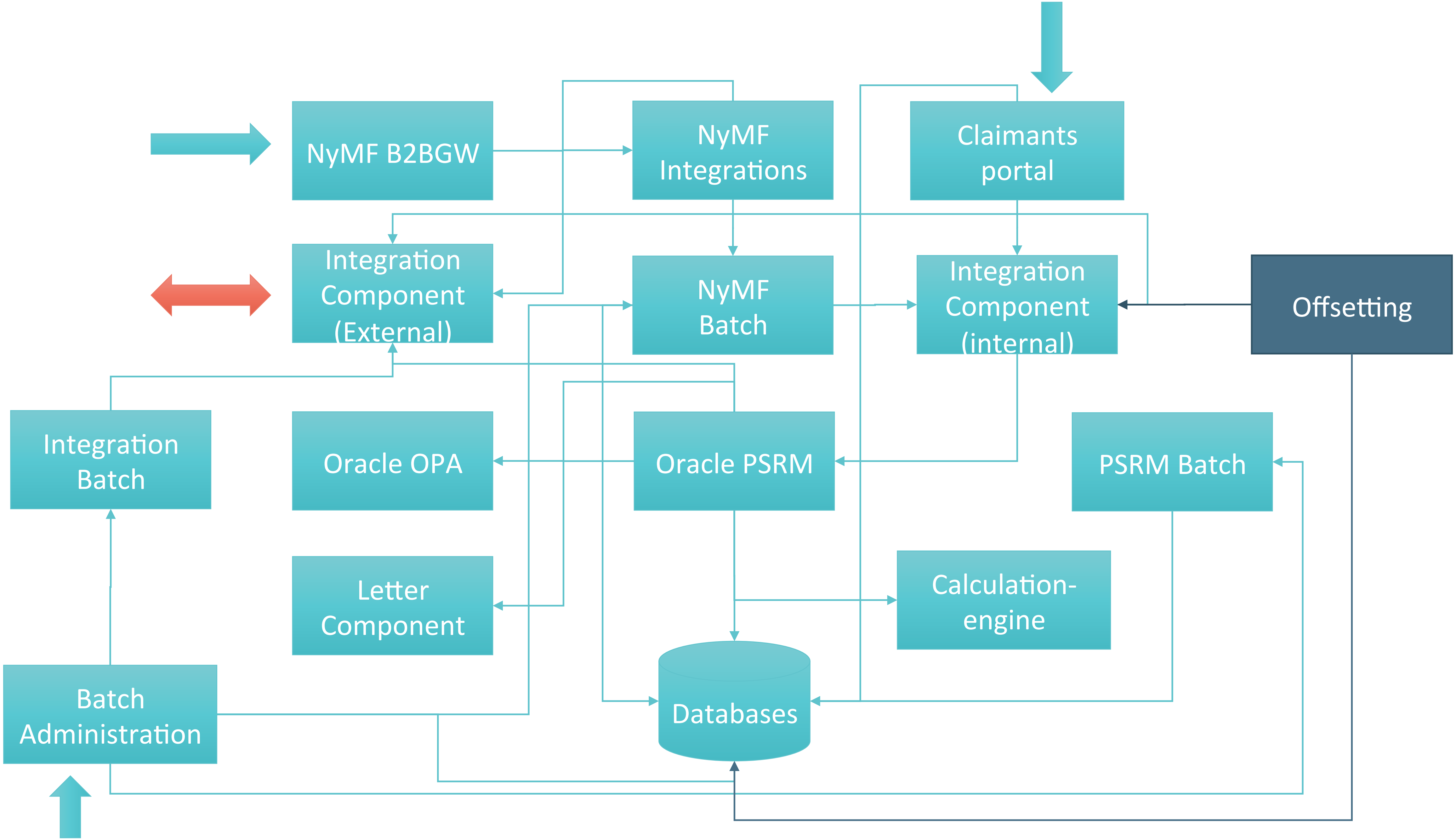
PSRM Batch

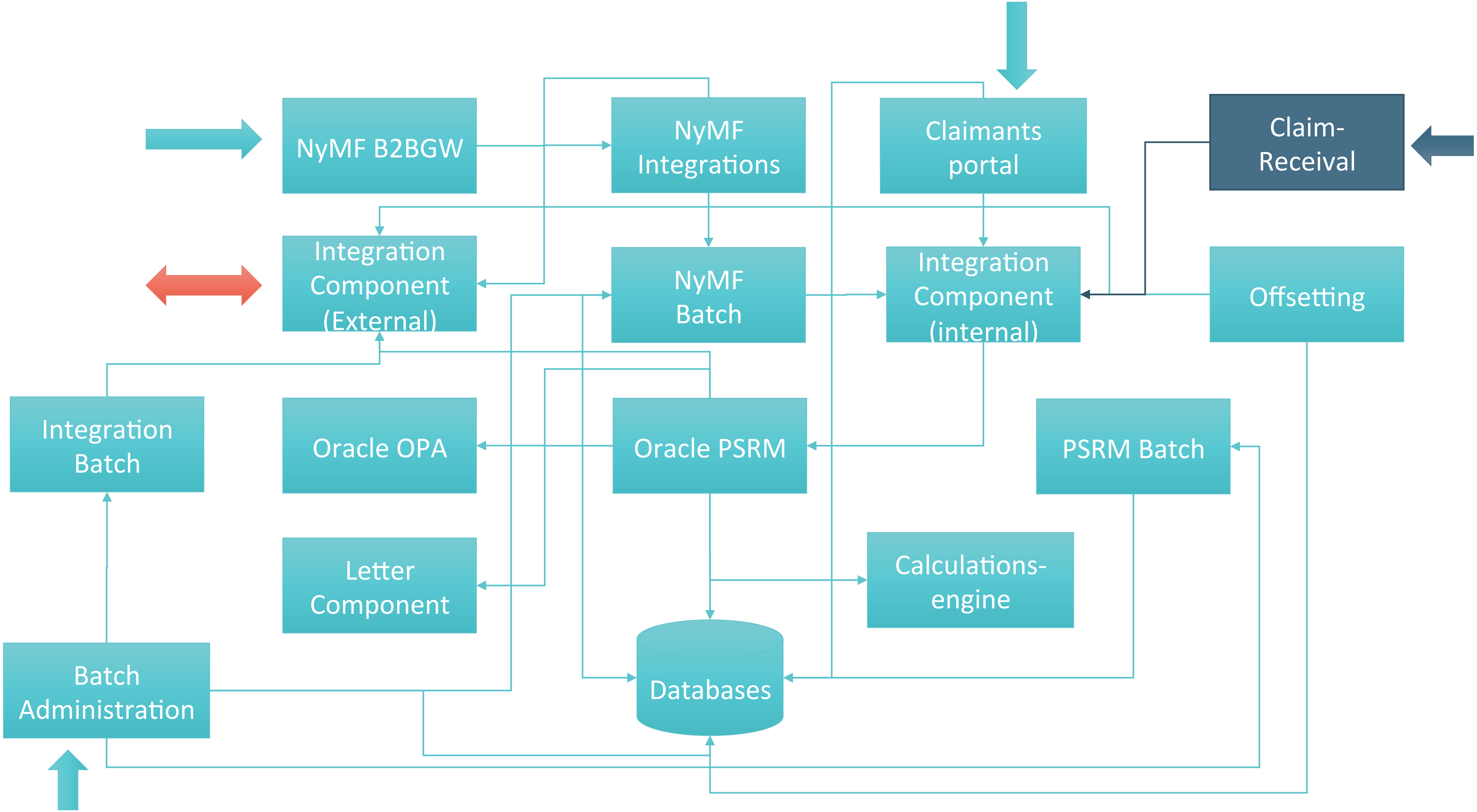
Letter Component

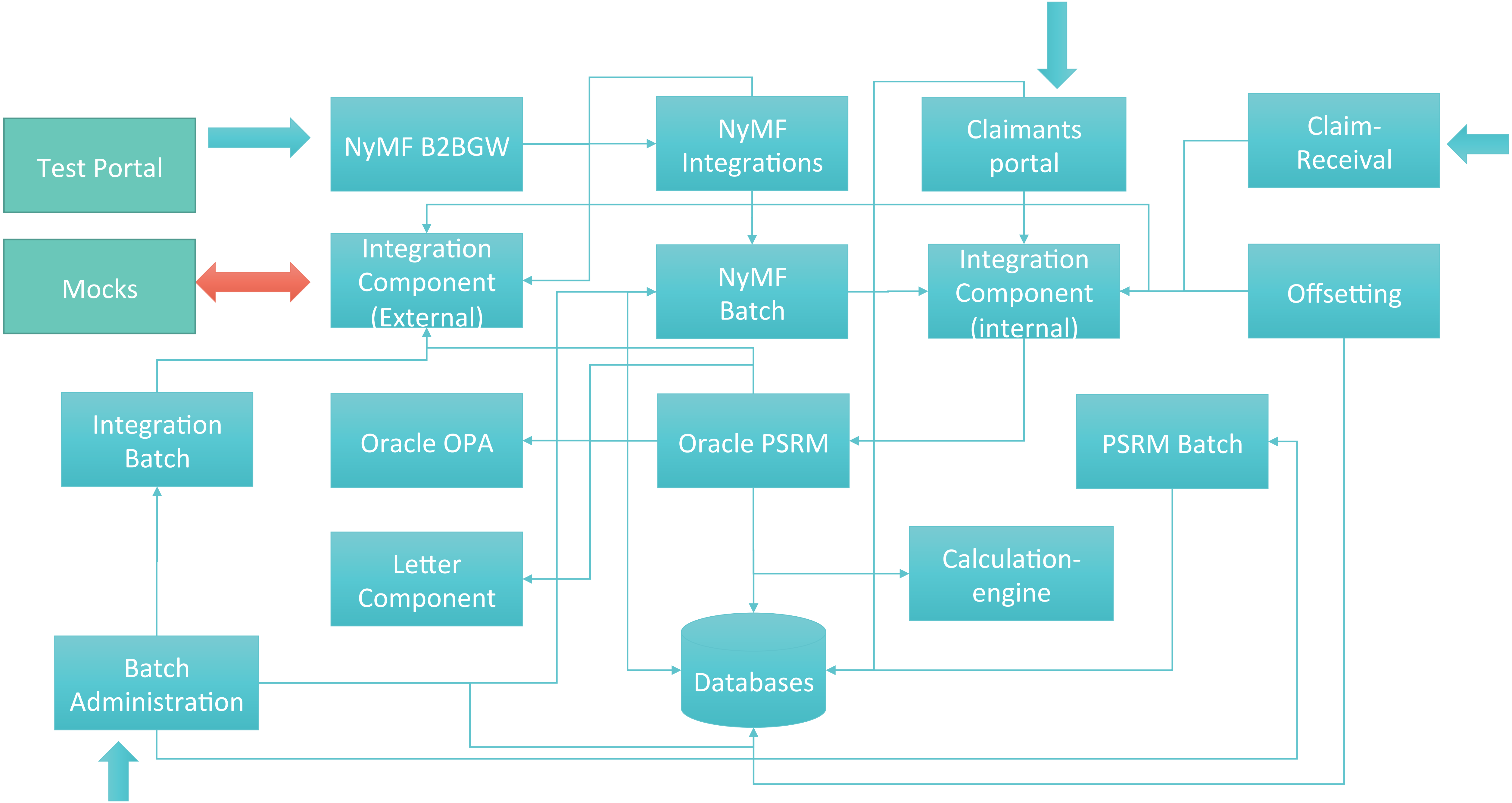
Calculation-engine

Databases

Batch Administration

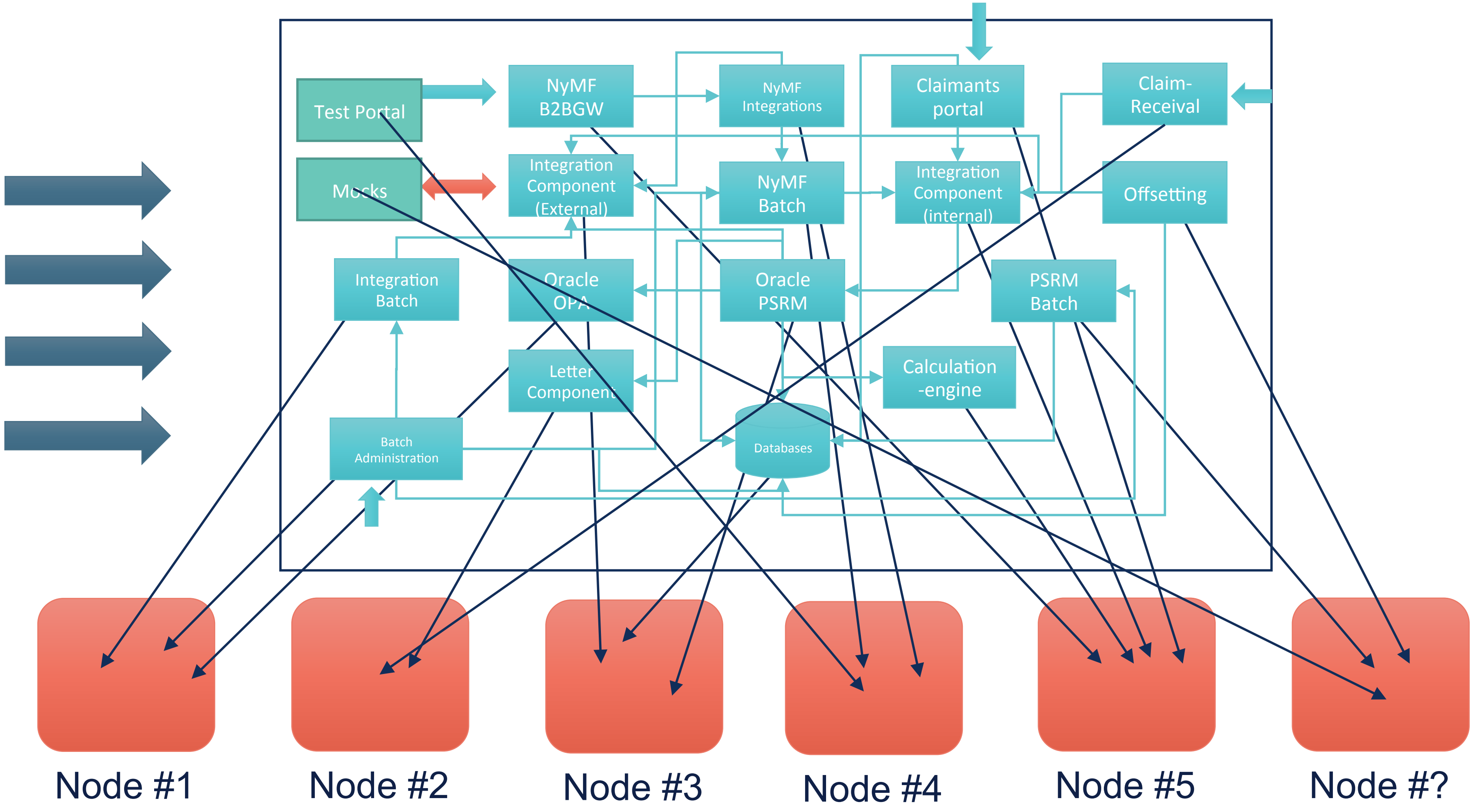

NyMF B2BGW
NyMF Integrations
Claimants portal
Integration Component (External)
NyMF Batch
Integration Component (internal)
Offsetting
Integration Batch
Oracle OPA
Oracle PSRM
PSRM Batch
Letter Component
Calculation-engine
Databases
Batch Administration


NyMF B2BGW
NyMF Integrations
Claimants portal
Claim-Receival
Integration Component (External)
NyMF Batch
Integration Component (internal)
Offsetting
Integration Batch
Oracle OPA
Oracle PSRM
PSRM Batch
Letter Component
Calculations-engine
Batch Administration
Databases


Test Portal
NyMF B2BGW
NyMF Integrations
Claimants portal
Claim-Receival
Mocks
Integration Component (External)
NyMF Batch
Integration Component (internal)
Offsetting
Integration Batch
Oracle OPA
Oracle PSRM
PSRM Batch
Letter Component
Calculation-engine
Batch Administration
Databases

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 Docker and 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 teammembers 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
    - Testdata
- 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