DEVCON 2018

Inclusive and inspiring topics from Alfresco community.

INSPIRATION AT WORK

How DevCon inspired Xenit, by Ronny Timmermans, CTO and Founder

XENIT' TALKS

Five talks from Xenit team, showing all our expertise, as Alfresco certified partner.

XENIT DIARY

xenit
return on content

ALFRESCO DEVCON 2018

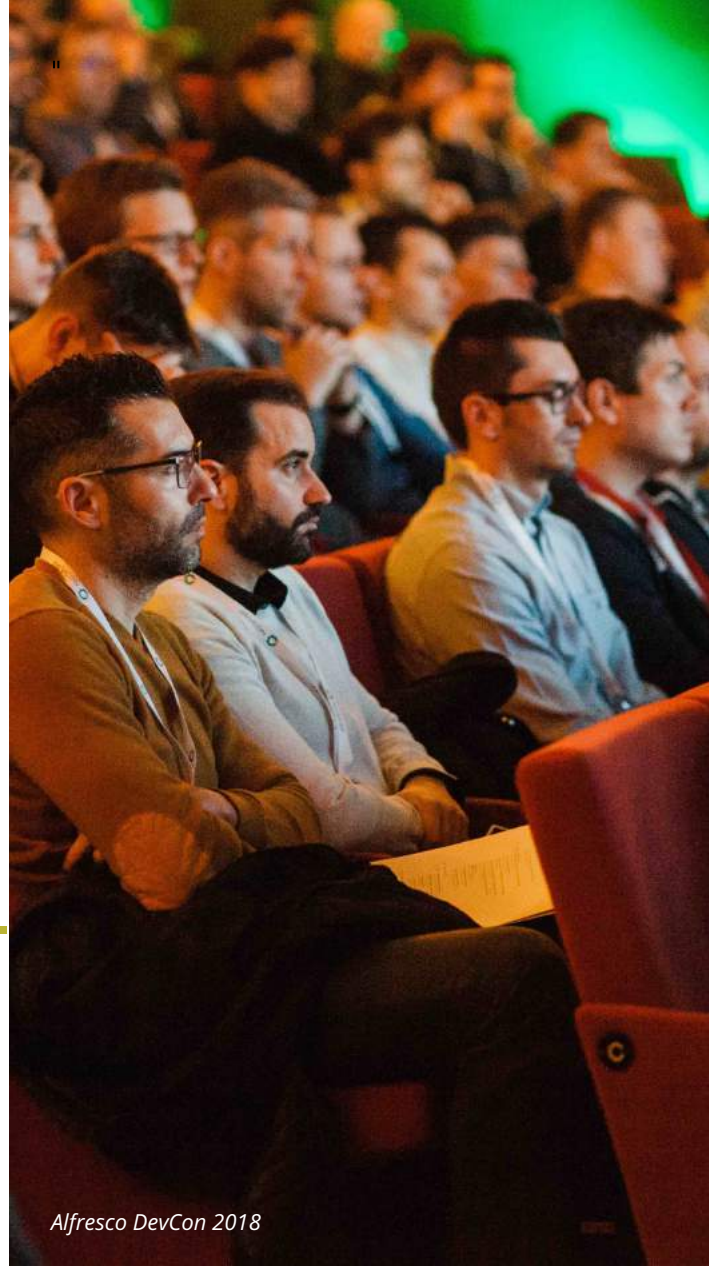
LISBON, PORTUGAL

JANUARY 16-18, 2018

DEVCON 2018 Lisboa

DEVCON IS BACK

Last month, developers, engineers, marketing and sales, customers, enthusiasts and evangelists, they were all together in Lisbon, Portugal, attending the Alfresco DevCon 2018.



Alfresco DevCon 2018

"VERY WELL BALANCED BETWEEN PURE TECHNICAL AND BROADER CONTENT"

LEARN. CONNECT. COLLABORATE

DevCon 2018 is an international developer conference entirely dedicated to Alfresco technology. With the support of the community, customers, and partners, DevCon, you will increase your technical know-how, connect you with other Alfresco developers, and let you collaborate with our team and each other.

SCAN THE QR CODE TO FIND ALL PRESENTATIONS FROM DEVCON

CLOUD IS THE NEW NORMAL

Production ready docker stackDevCon is not the first event where it has been mentioned, but now it's quite clear that the platform is preparing to move to the Cloud.
Amazon Web Service Cloud Service, precisely.

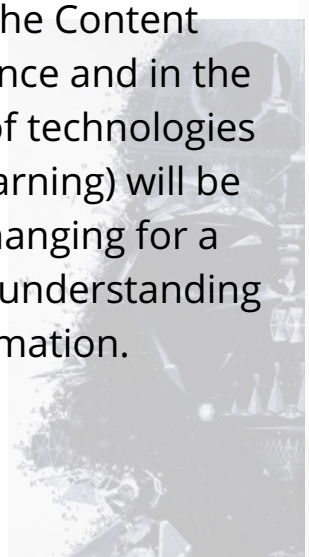
As Angel Borroy highlights in his blog post, "it seems clear from both talks, that Alfresco has started an isolation services process to prepare the platform to be deployed natively on Cloud. Probably, Activiti 7 (not available yet) will be the first product to be transformed in that direction, so we have to wait a little to start playing with these technologies."

WHO WILL BE THE NEXT JEDI?

You may be surprise of this association with Star Wars, but it was the theme used by John Newton, Alfresco CTO and Founder, to provide his vision on the next disruptive technologies.
In order:

- Artificial intelligence
- Microservices Architecture
- Blockchain
- Cloud operating System
- Edge Computing
- Multi-model User interface

The hearth of the Content Service is intelligence and in the next future a set of technologies (NLP, machine learning) will be one gaming – changing for a deeper and wider understanding of the information.



ADF AND THE FUTURE OF ALFRESCO SHARE

This topic created a lot of noise in the Alfresco community.

During the Q&A session, Thomas DeMeo, VP of the Product Management, answered a question about the future of Alfresco Share, saying the Share will be deprecated in the upcoming years, and ADF (APPLICATION DEVELOPMENT FRAMEWORK) will fill in a certain way this gap

In the Jeff Potts' article, you can read the effect of this decision and what might happen next...

INSPIRATION AT WORK

Ronny Timmermans
CTO and Founder @Xenit

Ronny Timmermans

Devon 2018 inspired Xenit to help re-inventing Content Services in next decade. "***ECM is not dead, Jim but it is transforming***".

Cloud and A.I. are creating new ways to manage content and automate the processes around it.

The cloud creates and advances future application architectures, even on premise, and monoliths are even more outdated than Dinosaurs

"Xenit seeks to develop these best practices, functionality and automation into our Alfred product line, with Alfresco technology at its core".

We retained a number of striking topics at DevCon 2018, in line with our sense of purpose to bring Cognitive Clarity @Zero latency for different markets, verticals and horizontal challenges like GDPR:

1. Alfresco is heading towards a new micro-services architecture and Xenit actually has built a number of components to accelerate that journey (Alfred Edge & Inflow).
Think docker containers and kubernetes, even for small scale solutions.

2. In the future, a Gateway will be part of the Alfresco reference architecture. We are looking forward to build this with, for or around Alfresco.

3. AI has a bunch of use cases in content and process services. Edge computing, Cloud, Blockchain, multi-modal interfaces and related technologies will make their mark in future content and process services.

4. Multi-modal *cognitive* interfaces will be the new norm, application and business integration will be smart and dynamic

5. Graphql, as launched by Facebook, fitting within our react-stack, is a very interesting specification for query and modification (mutation) and notification. It is used for interfacing with Activiti.

6. Alfresco is thinking of using Electron for packaging ADF applications on a desktop.

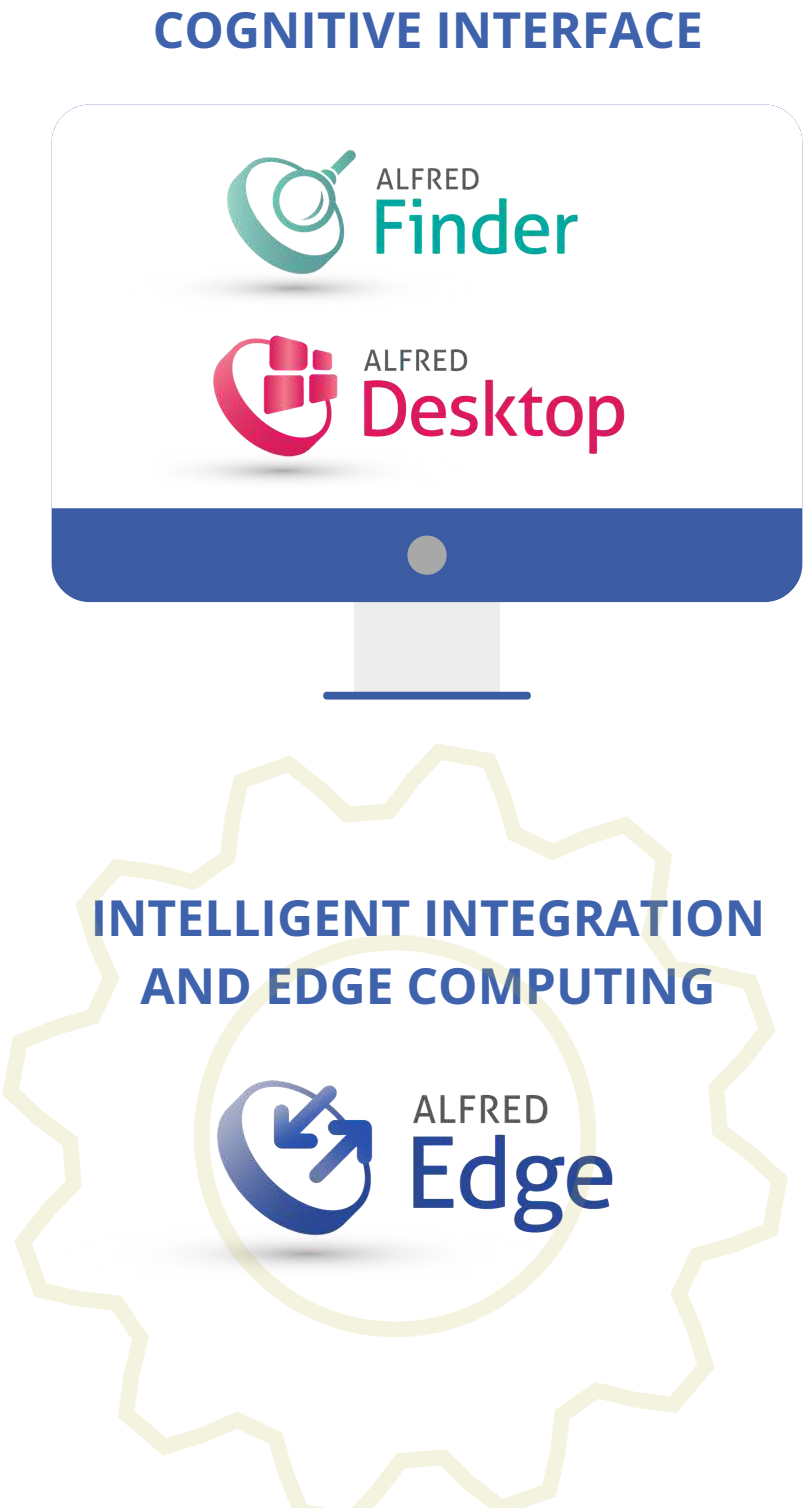
Ronny Timmermans

At Xenit, we think we can help Alfresco and its partners advance on this journey by :

1. Creating a community around Facebook's react and extending the ADF reach into the react community,

2. Extending our intelligent gateway Alfred Edge for synchronous and asynchronous communication, while infusing best 'edge computing' practices into it.

3. Continuing to deliver simple, smart and swift windows desktop experience with Alfred Desktop



XENIT TALKS

1. Roxana Angheluta
2. Thijs Lemmens
3. Jasper Hilven
4. Younes Regaieg
5. Willem Van den Eynde

PRODUCTION READY DOCKER STACK

1. Roxana Angheluta
2. Thijs Lemmens
3. Jasper Hilven
4. Younes Regaieg
5. Willem Van den Eynde

Roxana Angheluta: "**Production ready Docker stack**"

THE CHALLENGES

Automate the process of installing and maintaining Alfresco.

Have Alfresco image(s) prepared ready, both for the Community and Enterprise, with or without Solr and LibreOffice running inside, for clustered environments as well as non-clustered, remotely or on our hosted platform - reusing services such as the smtp server, with easy configuration changes and easy upgrades.

Keep track of all changes involved and install a full monitoring, which alerts in case of problems and helps in capacity planning for resources.

Flavor (community vs enterprise)	Single: Bundled / separated	Clustered	Hosted
Automation	Traceability	Versioning	Single deployment method
Upgrades	Easy configuration changes	Data separation	Security
Monitoring	Alerting		

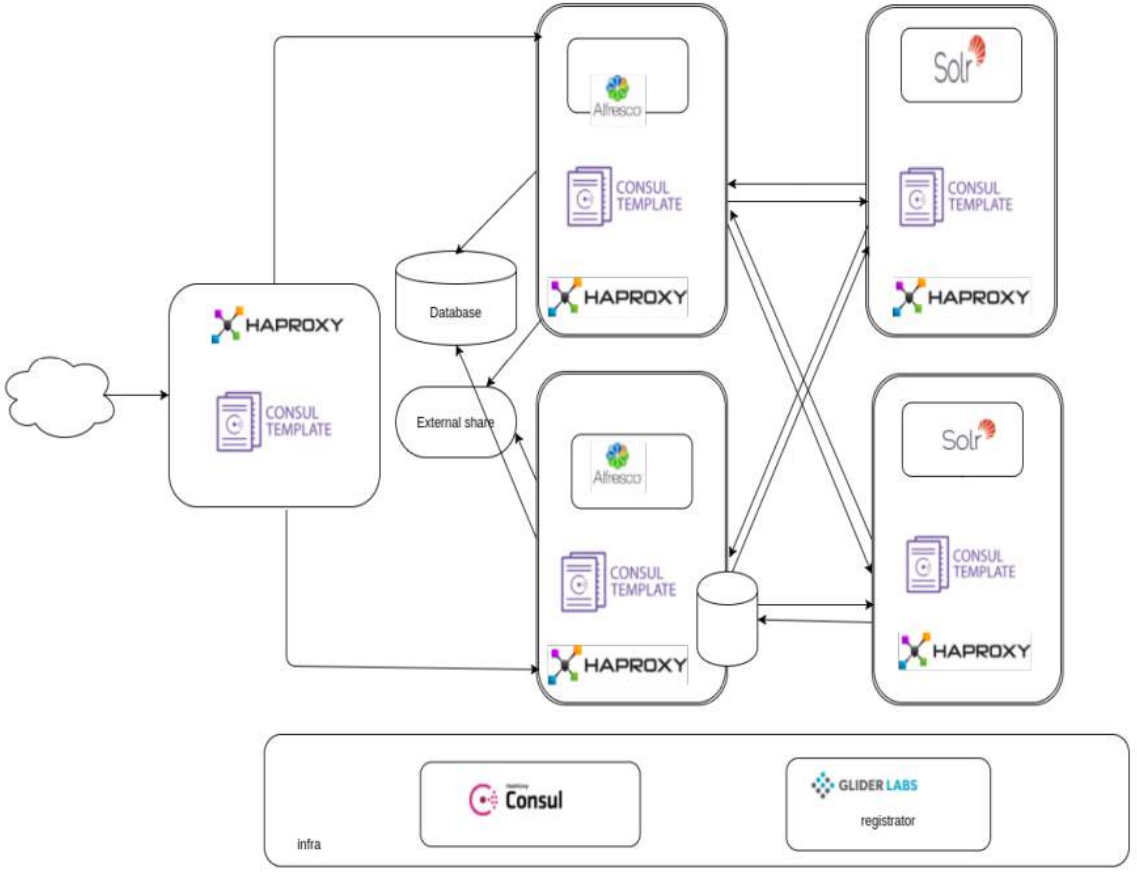
Roxana Angheluta: "**Production ready Docker stack**"

THE ARCHITECTURE

1. Docker-compose up/down to start / stop services.

2. Consul (Hashicorp) + registrator (GliderLabs) for service discovery.

3. Consul template (Hashicorp) for rendering.



"We started simple, with docker-compose files to start and stop services. We use Consul and registrator for service discovery and Consul template for rendering load balancing configurations and restarting load balancer. Consul was developed by Hashicorp and is a key-value store for storing services, adding or removing them on the fly based on health checks. The actual registration is done by registrator developed at GliderLabs." - ***Roxana Angheluta, Senior ECM Engineer @Xenit***

We already deployed this architecture to our clients, with 100% successful use cases

THE DOCKER IMAGES

The team started running the installer on a base Ubuntu image, very simple to implement and to upgrade to higher Alfresco versions.

When more flexibility is needed, like a specific version of java / tomcat, Installer is not right choice. The team moved therefore to a different method to build images.

As base layer they started from java, on top of which they added Tomcat, then a skeleton for an Alfresco including the init script, then the war files. Finally, client specific modules were added on top.

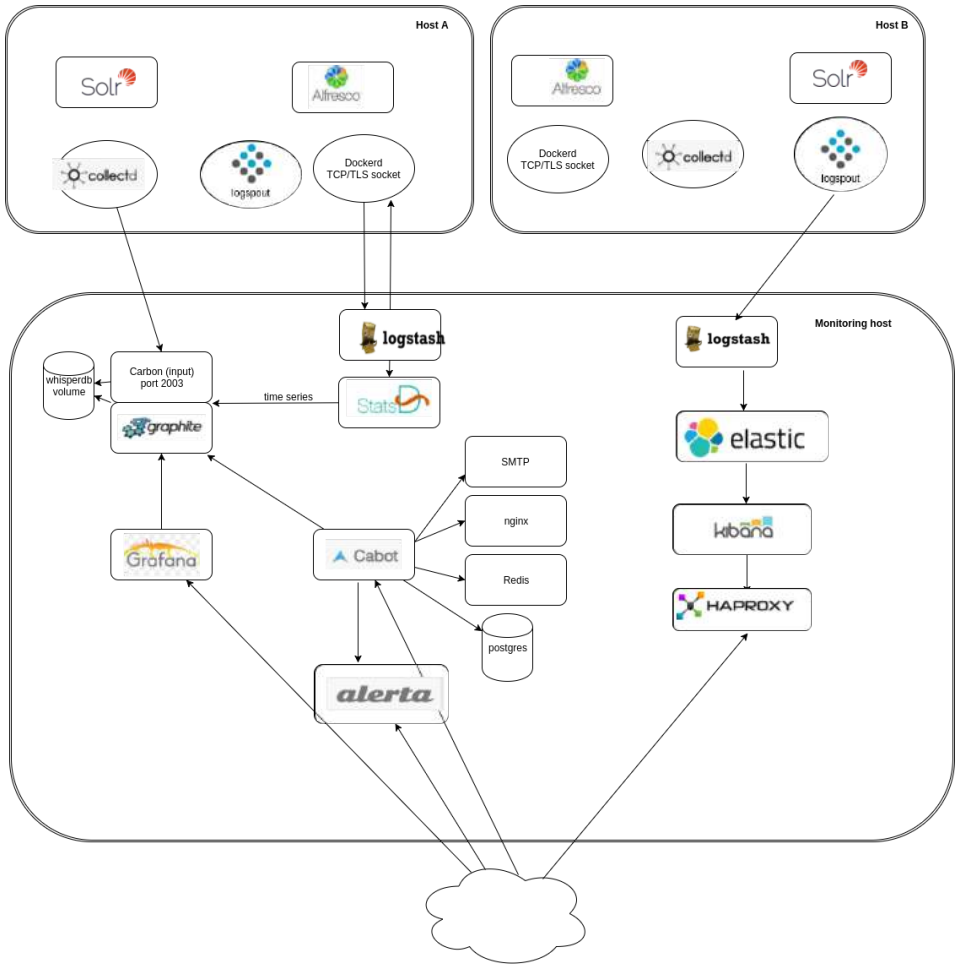
We aim for flexibility in customizing. We make the container stateless, by mounting the data and the logs.

Roxana Angheluta: "**Production ready Docker stack**"

THE MONITORING SYSTEM

Our monitoring stack is complex and includes a number of parts:

- Kibana, for log monitoring
- Grafana, for numerical data
- Cabot for alerting
- Alerta which aggregates Cabot warnings and alerts and allows you to spot and prioritize issues, across multiple clients.



What's next? Our next challenge is using reporting data to automatically produce SLA (Service Level Agreement) reports.

PostgreSQL FOR ALFRESCO: THE PRACTICAL GUIDE

1. Roxana Angheluta
2. Thijs Lemmens
3. Jasper Hilven
4. Willem Van den Eynde
5. Younes Regaieg

Thijs Lemmens: "**PostgreSQL for Alfresco: the practical guide**"

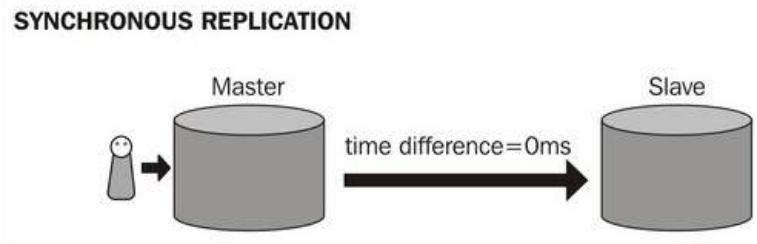
During DevCon, Thijs Lemmens, Senior ECM Engineer @Xenit, focused on three PostgreSQL topics:

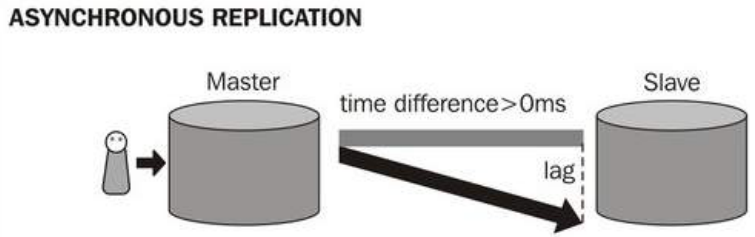
1. Master -Slave replication
2. Continuous backup and Point-in-time recovery
3. Upgrading database without downtime

Thijs Lemmens: "**PostgreSQL for Alfresco: the practical guide**"

In PostgreSQL, a master-slaves setup relies on a concept called streaming replication. This concept relies on the transaction logs (WAL) that ensure ACID compliance and copies those over from master to slave,or standby.

"Synchronous or Asynchronous replication. That is the question"





The advantage of synchronous replication is that when the master returns from a commit, you are sure that the commit is also safely written to the slave. In case of a disaster on the master, you don't loose any commits. However, when you want high availability, you need to have at least 2 slaves. Otherwise, the master will block writes, when the slave is down. Synchronous replication can have a large impact on performance, because it is highly dependent on the network latency, and throughput.

When using asynchronous replication, you don't have these drawbacks, but you have to take into account that there will be a replication lag. It is important to monitor this lag and alarm when the lag is higher than what you think is acceptable for your use case. In case of disaster, you might loose some transactions. This also means that you will have to restart Alfresco, because you risk having in consistent caches otherwise.

CONFIGURATION FOR DEMO PURPOSES

To configure PostgreSQL for streaming replication we have to adapt 2 files in the PostgreSQL data directory:

1. postgresql.conf - this is the main configuration file.
2. pg_hba.conf - In this file we can configure who as access to what database, from where.

MASTER	SLAVE	ALFRESCO

postgresql.conf:
wal_level = hot_standby # or higher
max_wal_senders = 20
 	
pg_hba.conf:
host replication repuser 172.18.0.0 255.255.255.0 md5
	Configuring the slave is very simple. You just need to run a command, and everything will be copied from the master, including the configuration
	
	pg_basebackup --pgdata=datadir --xlog-method=stream --write-recovery-conf --progress --dbname="host=pgmaster port=5432 user=repuser password=pwd"
		Alfresco uses a JDBC connection string. You can define multiple servers in a PostgreSQL JDBC connection string:
		
		jdbc:postgresql://pgmaster:5432,pgslave:5432/alfresco

CONTINUOUS BACKUP AND POINT-IN-TIME RECOVERY

Wal archiving concept: the wal files are copied to another location. The copied wal files can then be used when we want to restore a backup. This method has (at least) 3 advantages:

1. You don't need to schedule a nightly backup.
2. The backup is done continuously
3. Upon restore, you can decide to restore up until a certain time or transaction. If you made an error, like accidentally dropping a table, you can restore just before you made the mistake.

WAL ARCHIVING

To start the wal archiving we need to add the following lines in the postgresql.conf:

```
archive_mode = on
archive_command = 'test ! -f /wal_archive/%f && cp %p /wal_archive/%f'
```

The archive_command parameter uses 2 arguments:
- %f: The name of the wal file
- %p: The path of the wal file to copy

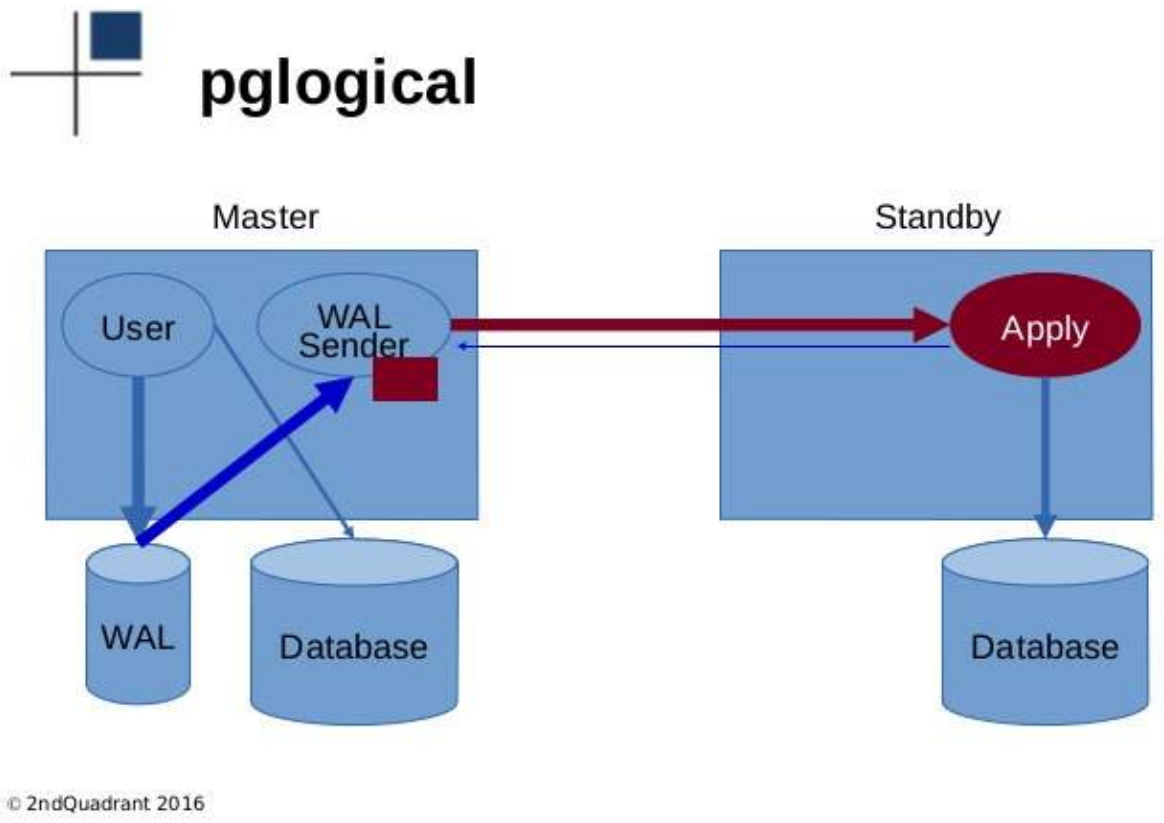
POINT-IN-TIME RECOVERY

Before starting the recovery, we need to restore the last base backup, taken before the time we want to recover. This base backup should become the new data directory. In that new data directory you need to put a recovery.conf:

```
restore_command = 'cp /wal_archive/%f %p'
recovery_target_time = '2018-01-09 09:39:50'
```

UPGRADING WITHOUT DOWNTIME

The easiest way to perform a database upgrade is using pg_dump to make a logical backup and to restore that backup on the new version of the database. The problem with this approach is that you need to perform these operations, and your database should be down, or in a read-only state.



A classic master-slave replication looks like viable solution. This way you could add a slave of the new version and switch over when all transactions are replicated.

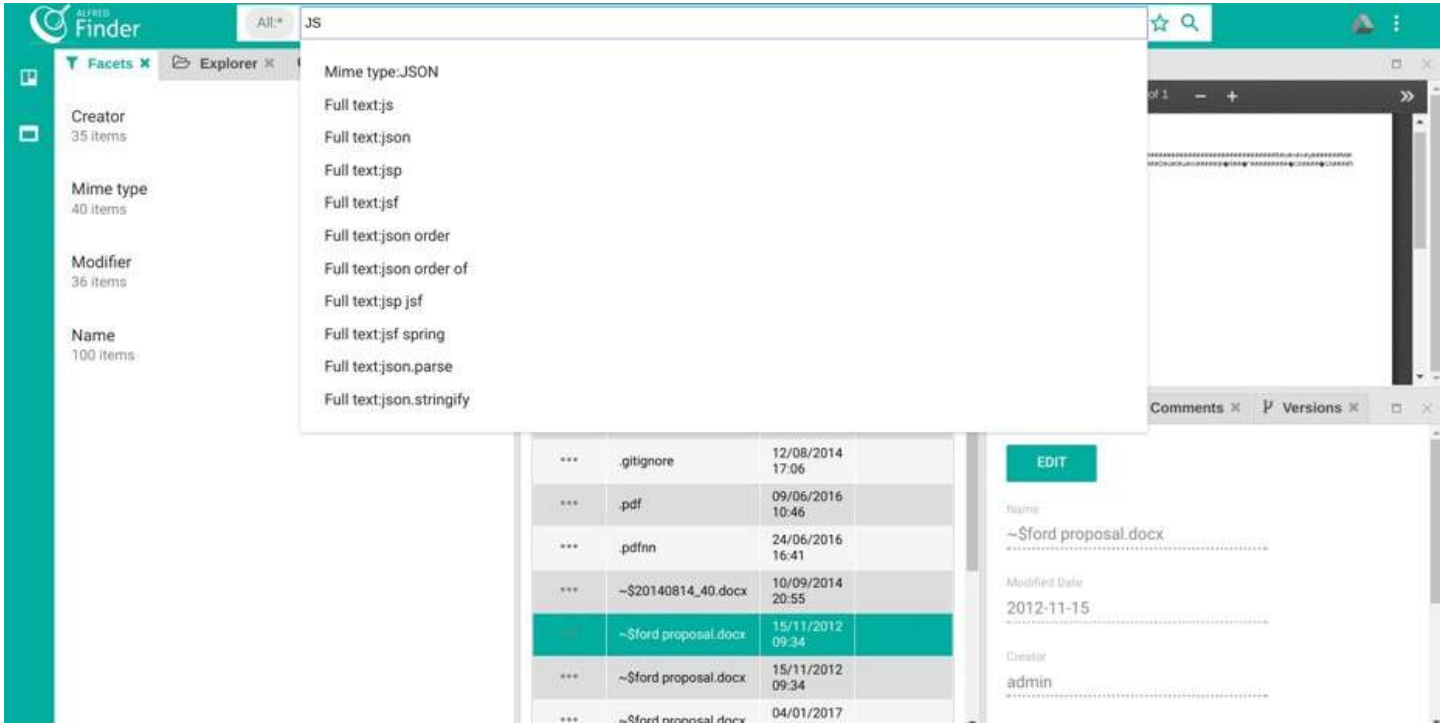
However, because the replication is based on a binary format, that changes between version, you can only use physical streaming replication between the same versions of PostgreSQL.
With the help of an extension **pglogical**, we can overcome this problem. Pglogical translates the physical statements to a version independent format, so they can be applied on another version of PostgreSQL.

HOW TO GET THINGS DONE FAST!

1. Roxana Angheluta
2. Thijs Lemmens
3. Jasper Hilven
4. Younes Regaieg
5. Willem Van den Eynde

Jasper Hilven: "How to get things done fast"

BEHIND THE SCENES OF ALFRED FINDER



Alfred Finder is a web application that allows you to quickly find documents on an Alfresco back-end, preview them and process metadata.
When customers use a software, most of the time, they don't think about the choices that are behind the development of it.

In this talk, **Jasper Hilven, Software Engineer @Xenit**, focused on two many aspects of building software:

1. **Managing complexity** and
2. **Ensuring quality at multiple levels.**

Jasper Hilven: "How to get things done fast"

MANAGING COMPLEXITY TO KEEP THINGS SIMPLE

In progress

Using React is a logic consequence of keeping things simple

React gives you the power of easily building the UI of your application without doing other complex things that we do not need.
It just works

Review

As software is build out of components, keep our components as simple as possible.

This implies that the complexity of a software component should grow naturally. This means you start with a simple value.

As soon as you want to reuse the value, you make a variable. If your variable needs a scope, you build a function or even later a separate class.

Done

We have multiple projects to ensure the right granularity of reuse, but also because we have multiple customers.

Customers are also not happy if you deploy their custom code to other customers.

We use NPM to manage the dependencies between these projects.

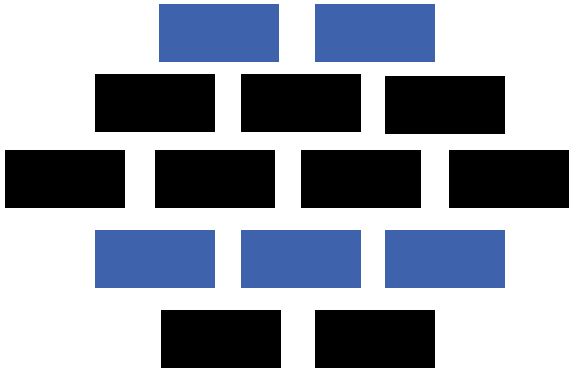
Jasper Hilven: "How to get things done fast"

ENSURE QUALITY AT MULTIPLE LEVELS

You need to find a balance between getting things done and assuring quality.

Quality of the code

@Xenit, we use **Typescript** to typecheck each single line of code. The cost of these checks are rather low, and happen each time you write a single line of code. We also use a Linter for style checking. Its called Typescript Linter. This is very useful if you have multiple people working on the same project.



Quality of the components

Components are the building block of software. We use unit tests to test our components. As soon as somebody breaks a component, a test complains about it. Then we can fix the issue long before the customer notices anything. "Its like having a machine that checks all the bricks each time again and again, before you are even building your house."

Quality of checks

We automatically run (almost) everything a user can do, each time we change something to product. This ensures that the customer can do with our software what we promise. Jenkins runs all these tests and keeps all the reports of all the performed tests.

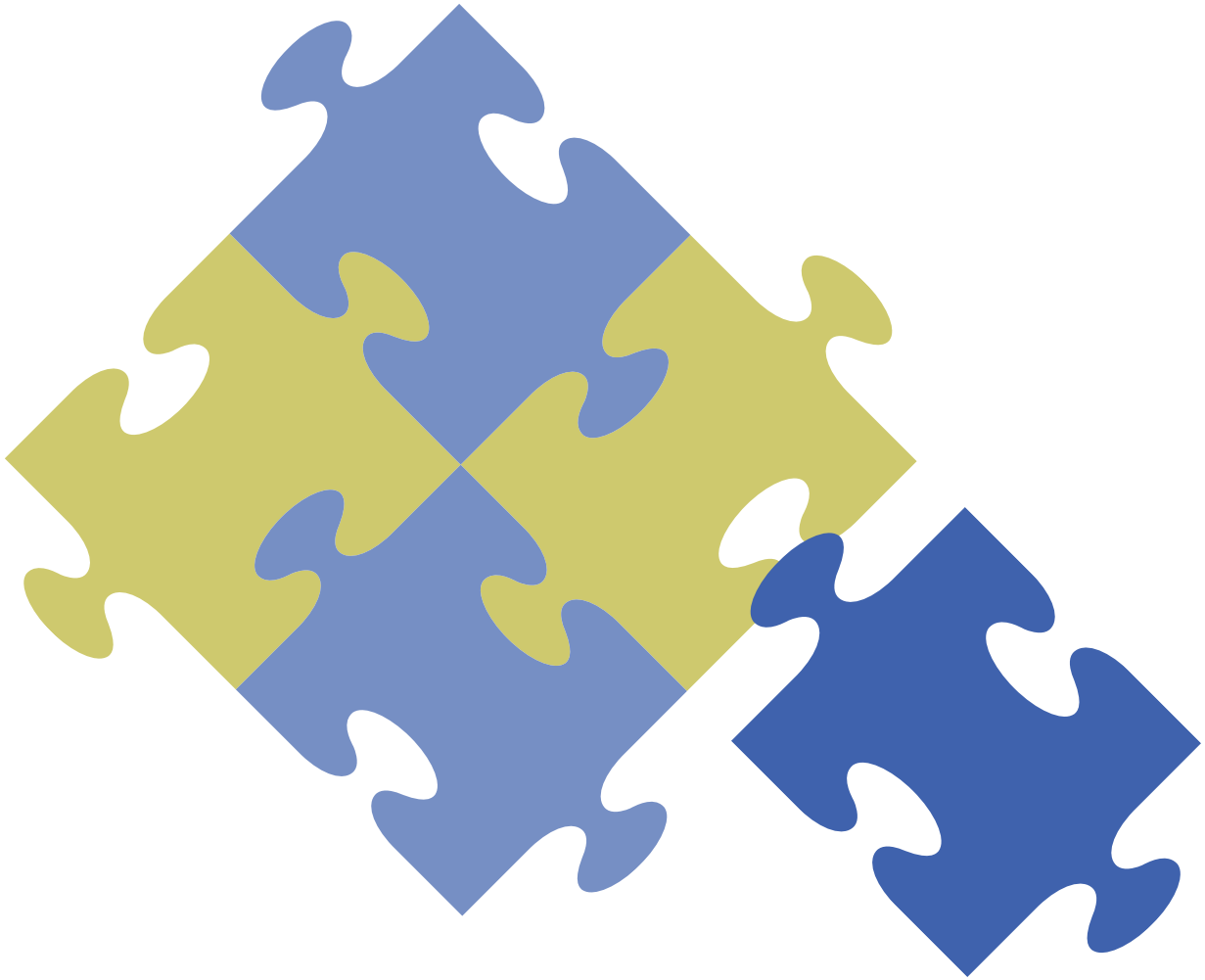
DON'T REINVENT THE WHEEL. REUSE!

1. Roxana Angheluta
2. Thijs Lemmens
3. Jasper Hilven
4. Younes Regaieg
5. Willem Van den Eynde

Younes Regaieg: "Don't reinvent the wheel. Reuse!"

Most software / systems are actually designed by orchestrating other smaller components (or bricks) that have been crafted or at least used in a different context.

Software engineering was initially more focused on original development but now it is recognized that to achieve better software faster and more efficiently, a design approach based on systematic reuse is actually needed.



"Actually reuse of existing assets in the software industry is not as new as some might think. Even though this industry started by promoting original software development, it definitely recognized that, in order to reach a better, faster and prone proof software, one would need to make use of reusable assets and directs most of the engineering efforts towards making the missing pieces in the puzzle, or maybe enhancing existing ones in order for them to fit perfectly in their environment."

-Younes Regaieg, Senior Software Engineer-

Younes Regaieg: "Don't reinvent the wheel. Reuse!"

WHAT CAN BE REUSED?

1. SOLUTIONS: For instance, a bunch of docker container images / or VMs (Virtual Machines) to speed up the deployment process.

2. APPLICATIONS: By adopting Alfresco Content Services, a fully fledged applications are reused

3. COMPONENTS: A component can be actually be an add-on or a library that bundles a feature or set of tightly coupled/related features.

4.OBJECTS AND FUNCTIONS: Same as components, they represent a piece of reusable logic, but at a smaller scale.

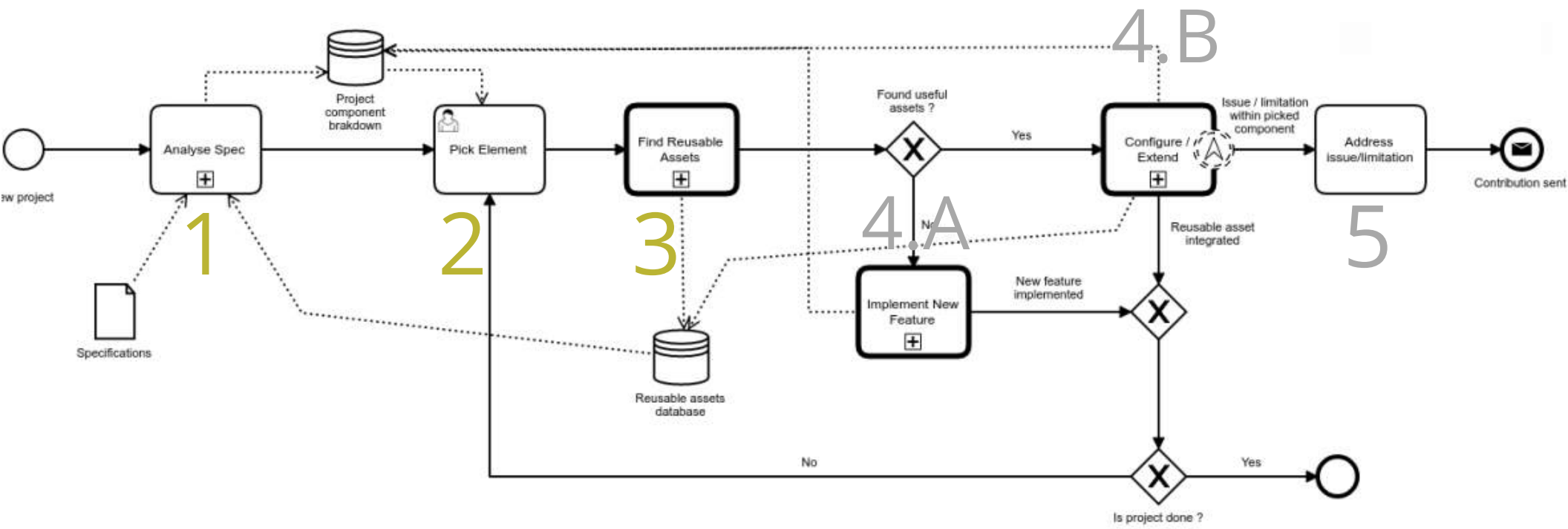
Younes Regaieg: "Don't reinvent the wheel. Reuse!"

A SUSTAINABLE SOFTWARE REUSE PROCESS IN FIVE STEPS

1. **ANALYZE SPECIFICATIONS:** Usually at the start of a project you get the software specifications which you use to create a component breakdown to be addressed one piece at a time.

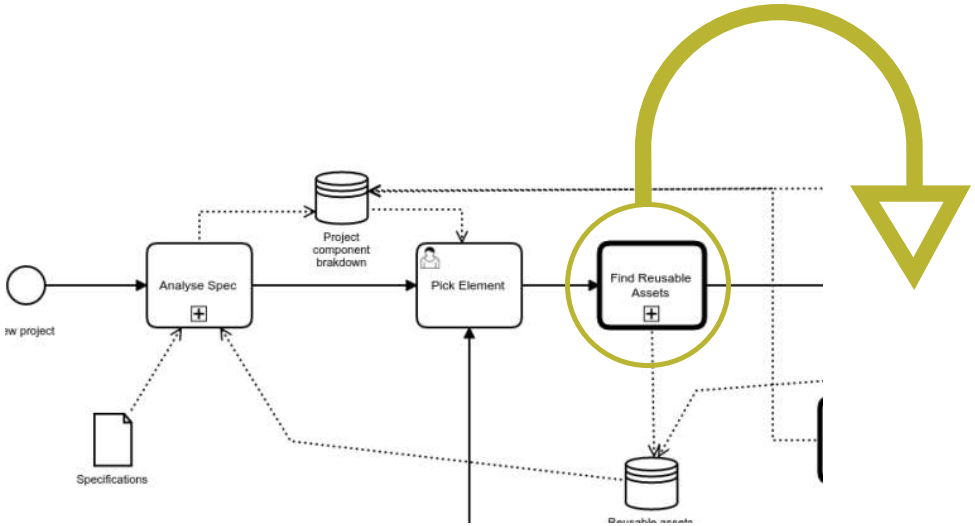
2. **PICK ELEMENT**: Then you start picking pieces and address them one by one.

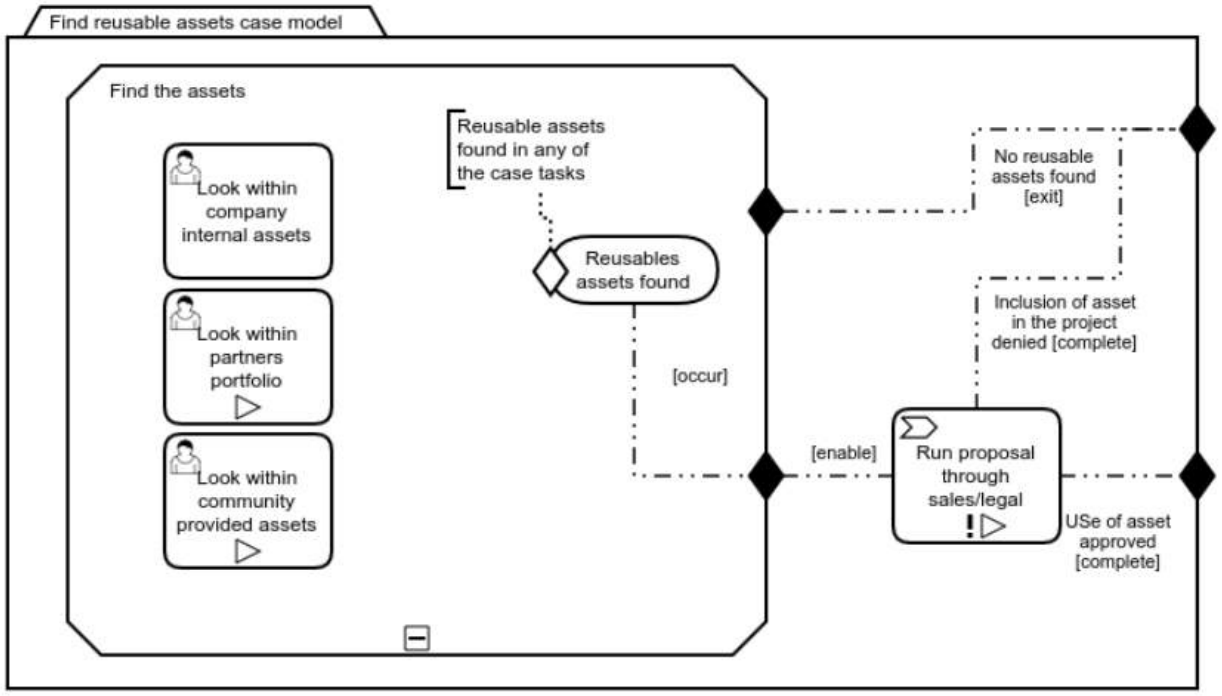
3. **FIND REUSABLE ASSETS**: This step would highly depend on the knowledge of the team on board as well as information that is available in the context of the project.



Younes Regaieg: "Don't reinvent the wheel. Reuse!"

FIND REUSABLE ASSETS USE CASE





In this use case, we are looking for the company internal assets catalogs for reusable assets and optionally looking within partners portfolios and community projects.
This depends on the customer in question, the required support level, the available budget and the experience of the team members involved.
There are two options:

First option: Not finding any assets that could be useful, exit and terminate the activity

Second option: Find assets and run them though legal/sales departments

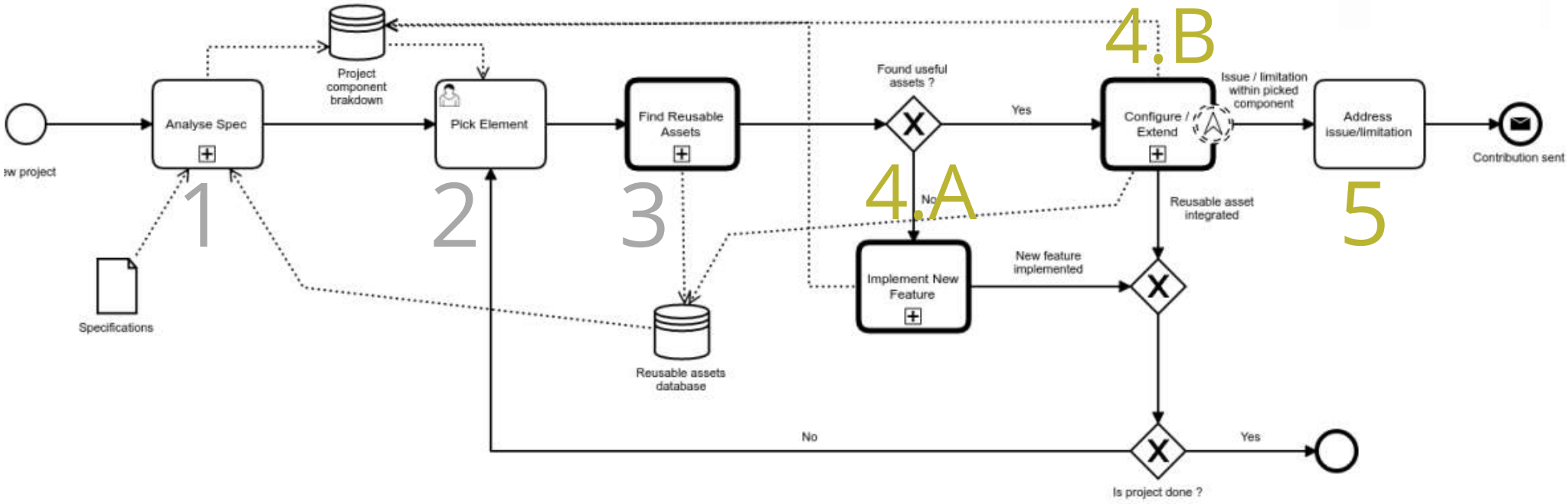
Younes Regaieg: "Don't reinvent the wheel. Reuse!"

A SUSTAINABLE SOFTWARE REUSE PROCESS IN FIVE STEPS

4.A **IMPLEMENT NEW FEATURES:** If you didn't find any assets, fit for the reuse, then you can implement the new features

4.B. **CONFIGURE AND EXTEND**: Our search might have been a success and so we go forward to the step where we configure or extend (or even implement something on top of it) the found re-usable assets.

5. **MANAGE POTENTIAL ISSUES:** You might eventually land on an asset where we can not cover all requirements out of the box or we encounter a bug that needs resolving. in such case you might escalate to a new path where patch the reusable asset and eventually send out a contribution for the project in question.



Following a couple of iterations on this process, your organization will start to develop a knowledge base for reusable addons, with an increased productivity, faster time to market and definitely a better software quality

3 TIPS TO MIGRATE ALFRESCO TO AWS OR AZURE

1. Roxana Angheluta
2. Thijs Lemmens
3. Jasper Hilven
4. Younes Regaieg
5. Willem Van den Eynde

Willem Van den Eynde " **3 tips to migrate Alfresco to AWS or Azure**

Microsoft Azure

amazon web services

Cloud infrastructure offers many benefits including cost-savings and transparency in architecture. **Willem Van den Eynde, Senior ECM Sofwtare Architect @Xenit,** got in touch with AWS services in 2012.

However client adoptions was lacking behind. Until 2016, when he did two migrations from an existing on-premise installation to AWS and a similar one to Microsoft Azure.

Willem Van den Eynde " **3 tips to migrate Alfresco to AWS or Azure**

Microsoft Azure

1

amazon web services

TIP #01 - COPY DATA FIRST: Through the 2 migration projects, copying over the data has been the bottleneck. It's often slow to copy over lots of small files. To give an example: it takes about 15 days to copy over 4TB over a 3MPBps connection. Speeding up the process by using temporary storage solutions and copy over deltas also adds to the overall project throughput.

Willem Van den Eynde " **3 tips to migrate Alfresco to AWS or Azure**

TIP #02 - CLOUD ARCHITECTURE: Cloud architectures are very transparent, because all information about their possibilities and prices are available online. This transparency helps to easy co-create the desired architecture between client, Alfresco partner and infrastructure partner.
Components are regularly added and updated to take into account the latest technology and are easy to evaluate and access.

.

Willem Van den Eynde " **3 tips to migrate Alfresco to AWS or Azure**

TIP #03 - MIGRATE FIRST, THEN...: It's a migration project first. Those projects typically require a very short cut-off period, during the weekend for example. Therefore it's best to limit the scope to just the migration. And leave other ideas like upgrades to subsequent projects.

Thank you