

Luther Case Study

Cross-Department Claims Settlement end-to-end Process Operations



Allianz 

 **Luther
Systems**

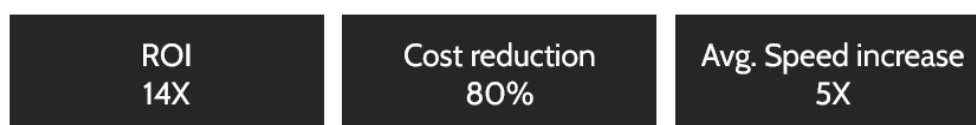
1. Executive Summary

Allianz SE is one of the largest financial services companies in the world, employing over 157,000 people and serving around 125 million customers worldwide¹. It reported an operating revenue of €161.7 billion in 2023¹. Allianz is also a key player in asset management, overseeing approximately €2.5 trillion in third-party assets through its subsidiaries, PIMCO and Allianz Global Investors². Allianz is also the largest insurer in the world. Allianz issues a significant number of insurance policies, particularly excelling in motor insurance. Globally, Allianz collected over €2 trillion in property and casualty insurance premiums, which includes motor insurance³.

Allianz provides insurance in a variety of categories, including life & health insurance, property & casualty insurance, corporate & specialty insurance, and credit insurance. Motor insurance makes up part of property & casualty insurance. Allianz underwrite motor insurance policies that cover damage to the policyholder's vehicle, as well as damage caused to other vehicles in accidents and to the property of others. Because vehicles can travel across Operating Entity jurisdiction borders, Allianz motor insurance policies are tied to the person and not the Operating Entity (OE) where the policy was issued. Therefore, motor insurance claims often need to be handled cross-department.

Allianz provides motor insurance policies to customers in 23 operating entities (OE) in Europe. Each OE operates siloed and independently, with their own teams. Many motor insurance customers will purchase a policy under one OE, and then later if they have an accident they may have to make a claim with a local OE. The entity that issued the policy is responsible for settling the claim, but the local entity is more well equipped to handle local regulations and services. Therefore, the local entity (the "Handling" entity) processes the claim while the entity that issued the policy (the "Originating" entity) funds the process. However, siloed and separate process operations lead to large amounts of manual reconciliation of different data formats and documents, which delays claims and increases operating costs.

The commercial impact of implementing the Luther Platform for Allianz are:



The process involves 5 teams: i) the Handling Claims Team, ii) the Originating Claims Team, iii) the Handling Finance Team, iv) the Originating Finance Team and v) the Originating Payments Team. A full breakdown of teams and their roles can be found in section 2.4.

A key component of claims processing is cross-department claims settlement. The cross-department claims settlement process involves several key steps: retrieving and processing the claim, validating the claim, generating an invoice for the originating entity, reviewing and approving the invoice, and making the payment. For a full breakdown of the process see section 2.5.

¹<https://www.allianz.com/en/about-us/company/at-a-glance.html>

²<https://www.cityam.com/allianz-under-investigation-by-german-regulator/>

³https://www.allianz.com/en/economic_research/insights/publications/specials_fmo/2024_05_23-Global-Insurance-R-report.html

Allianz handles 50,000 cross-department claims annually, and each cross-department claim awards an average of €4,000. This amounts to €200 million annually.

Annual cross-department claims 50,000	Value of all claims annually €200,000,000	Average value of each claim €4,000
--	--	---------------------------------------

Allianz operates the cross-department claims settlement process as part of the customer relations management value chain. This process operates across 5 teams & 8 software systems and it includes 156 tasks end-to-end.

Teams 5	Software systems 8	Tasks 156
------------	-----------------------	--------------

To operate the process end-to-end, each team operates a number of functions. Each function performs the same Operations Cycle (series of steps); i) send data & info to the system, ii) receive response from system, iii) compute & validate response, iv) share & store execution of step, v) evaluate & initiate next step.

Operational Silos cause unreliable Process Operations

Allianz operate the Process across 5 teams & 8 Software Systems

each team & system performs a function for the Process (perform checks, review docs, ...)

these teams & systems are siloed, they have separate ops, tech & governance

but the end-to-end process operates across them

For reliable operations, all teams & systems involved should operate the same end-to-end process. However, they often don't! This leads to operational & technical challenges, which make process operations unreliable. The opportunity is providing a platform to reliably operate the end-to-end process, across all teams & systems involved. Traditional solutions to end-to-end process operations are unreliable & expensive.

Enterprise Operations are generally function-first, which means they continue to focus on improving functions & systems, but processes are considered secondary. The thinking is that if we have great functions & systems, the business can operate any process! Traditionally enterprises use bespoke connectors & local operations scripts for process operations, which are fragmented, siloed, and changed separately, and so are ineffective for reliable process operations.

Enterprises primarily focus on the operations of individual teams & systems, and continuously improve them

operations of the end-to-end process across 5 teams & 8 systems is of secondary focus, especially as the process evolves

This costs the enterprise millions in operational costs, and days in delays

For Allianz's claims settlement process, this leads to

Practical Problems in the settlement process

Settlement delays lower customer satisfaction

Differing policy processing leads to inconsistent service

Lack of transparency leads to customer confusion

High outstanding debts impede operations

Difficulties in performing audits efficiently

Damage to Allianz's reputation across Europe

Inconsistent fraud detection

To remedy this, enterprises use automation tools. However, they are ineffective at end-to-end process operations, due to their limited scope and scale, and stitching them together also doesn't solve the problem.

Luther's platform is designed process-first, & primarily focuses on end-to-end processes. Reliable end-to-end process operations include consistent operations, and great functions & systems.

Luther's platform takes a Process First Approach

focusing on reliable operations of the end-to-end process across all teams & systems,
instead of cobbling & stitching together the separate & siloed functions
of 5 teams & 8 software systems

The automated cross-department claims settlement process, built on the Luther Platform and developed by Luther in collaboration with Allianz, uses Deep Process Automation Technology to improve the automation of claims settlement by separate Allianz entities in different companies, and automatically process and verify claim data on the Luther Platform. The Luther Platform provides standard connectivity and a Common Operations Script shared by all participants. The platform reliably operates the end-to-end process across all teams and software systems from the common operations script.

Luther's unique value for reliable end-to-end Process Operations is providing i) standard connectivity & ii) a common operations script, across all teams & software systems.

Luther's unique value for reliable end-to-end Process Operations is providing

standard connectivity
a common operations script

across all teams and software systems.

Luther's platform vertically integrates i) distributed system technology ii) optimal resource allocation & management, iii) real time event ordering & streaming, iv) deterministic event processing & execution, for reliable end-to-end process operations.



Luther's platform does this by i) connecting systems to standard platform nodes, rather than to each other, and ii) teams & systems can change the common operations script but all teams & systems have to know & agree to the change, so all teams & systems involved operate the same end-to-end Process all the time!

Finally the Luther Platform reliably operates the end-to-end cross-department claims settlement Process across 5 teams, 8 software systems & 156 tasks.

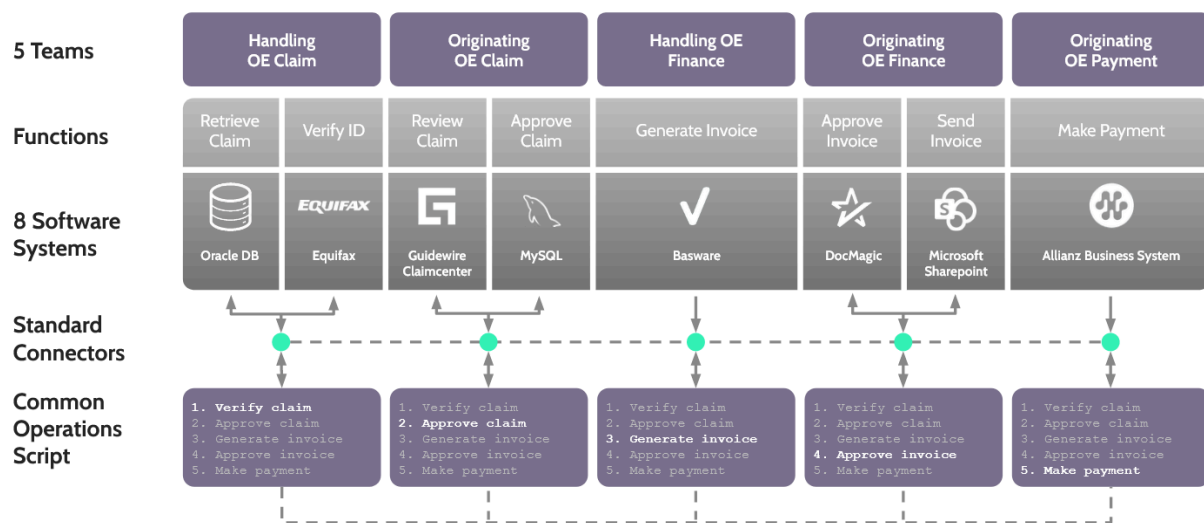


Fig 1. Cross-Department Claims Settlement Process operations built on the Luther Platform

To implement the platform, i) Luther's team mapped the Process, ii) Identified teams & software systems in the process, iii) allocated nodes (servers) to teams, iv) connected nodes to systems, v) set up the Platform on the nodes. vi) Allianz's team along with Luther's team developed the Common Operations Script (code) for Process Operations, vii) the process went Live.

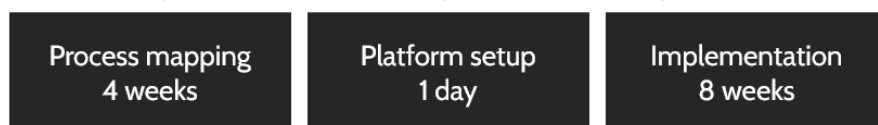


Fig 2. Implementation timeline of the automated Cross-Department Claims Settlement Process.

The results have been highly impactful. Thanks to increases in efficiency and operations reliability, a process that traditionally took 4 hours can now be completed in less than 0.8 hours and operational costs have been reduced by 80%. Beyond the commercial results, this led to operational benefits in production; i) reliable operations across the end-to-end process & over time, ii) 5X smaller Ops teams, iii) real-time monitoring, iv) enforced compliance checks, v) real-time and consistent updates across all teams, vi) reduction of reconciliation. Also, technical benefits during development; i) standard dev. process so developers can focus on operations, ii) 5X smaller Dev teams, iii) standard process connectors, iv) automated infrastructure and connectors setup, v) real-time and consistent updates with the rest of operations.

Operational benefits in production	Technical benefits during development
Consistent Ops: across the process	Standard development process
Consistent Ops: over time	Devs focus only on development of process Ops
5X smaller Ops teams	Eliminate DevOps teams, 5X smaller Dev teams
Real-time monitoring	Consistent, standard process connectors
Enforced compliance checks	Consistent process Ops scripts across teams
Real-time & consistent updates across teams	Standard infra., connectors, dev. env. setup
Reduction of reconciliation	Real-time & consistent updates across operations

The key impact is the increased efficiency of the cross-department claims settlement process. The Luther Platform standardizes data entry formats and eliminates the need for manual checks of documents and time intensive manual reconciliation, reducing operating costs and times. Customers are compensated faster, increasing customer satisfaction while reducing operating costs for Allianz. The platform could be further expanded to include all operating entities that Allianz control, and the Luther platform could also be utilized to further streamline other areas of Allianz's operations, for example using automation to streamline other areas of the motor insurance value chain, such as repairs, motor purchase, and policy issuance.

ROI 14X	Cost reduction 80%	Avg. Speed increase 5X
------------	-----------------------	---------------------------

Fig 3. Results of implementing the Platform for the Cross-Department Claims Settlement Process



2. The Process

2.1. Process Operations

Different teams have different operations, rules and governance and they also utilize and operate a variety of software systems in different ways. Each system operates a specific function for the process. To operate the process end-to-end, each function performs the same cycle of steps: i) **Send**: send data & information to the System, ii) **Receive**: receive response from the System, iii) **Validate**: compute & validate response, iv) **Store**: share & store execution of step, v) **Initiate**: evaluate & initiate next steps.

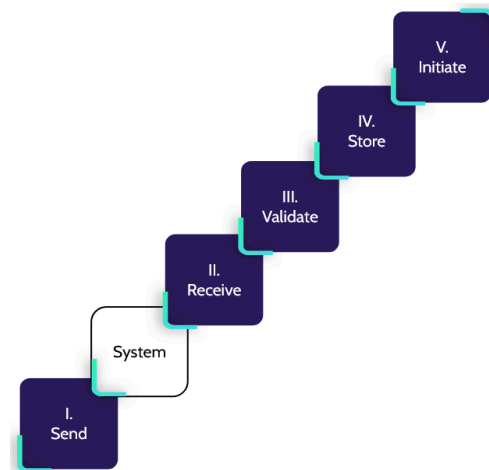


Fig 4. These are the requirements that repeat for all functions across the end-to-end Process Operations.

Enterprises operate a set of specific functions based on their objective. For example, an investment management company's functions help it to manage investment portfolios. While the functions and systems may change, the process remains the same. However, expecting processes to be efficient because of efficient individual tools simply does not work for enterprises. Luther empowers enterprises with a process-first approach.

Tasks are simple events that are localized to one team involving one or two software systems, for example retrieving data from a database. Workflows are more complex, involving 10-20 tasks between one to two teams, and two to three software systems. An example of a workflow is onboarding a new employee. Processes are complex, involving 50+ tasks, 3 or more teams and multiple software systems. Settling a claim end-to-end is a process.

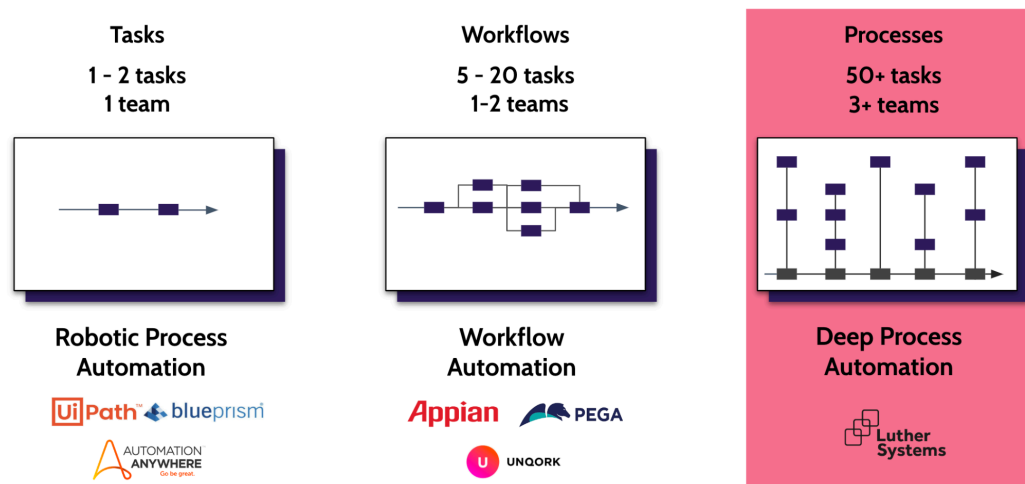


Fig 5. Different tools are used to automate different levels of complexity.

2.2. Function First Operations and its limitations

Generally, enterprise operations are function-driven. They have a large collection of software systems each operating a specific task. Tasks often have dedicated software systems and are operated by specific teams. By developing, purchasing and maintaining efficient systems, most enterprise tasks operate highly reliably.

Enterprise processes, however, operate across multiple teams and software systems, and involve many tasks. This means reliable end-to-end process operations require efficient teams and systems, as well as efficient connectivity and operations across these teams and systems.

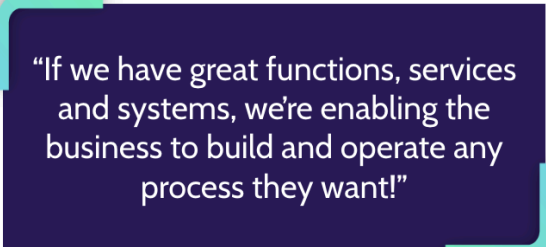
Enterprises generally take a “function-first” approach to process operations. Great individual teams and systems provide the required ingredients for great process operations, so they focus on enhancing and improving the performance and efficiency of individual teams and software systems. A good analogy of this approach is “if we have great ingredients, anyone can cook anything they want and it’ll be of great quality!” Processes are considered secondary to functions and systems, as they are considered ever changing, and efficient functions and systems can enable any process that the business may envision. The problem is, efficient functions do not necessarily create an efficient process.

Efficient software systems and functions are not enough to reliably operate a process end-to-end

In practice, most enterprises have a defined charter and mission, particularly if they are in a regulated industry. They provide specific products and services which are generally enumerated and these rarely change. These form the basis of the value streams provided by an enterprise. For example, every insurance company assesses risks, underwrites policies, manages premiums, processes claims, and adjusts or terminates coverage. The majority of “enterprise operations” are in operating these value streams. Each value stream has a set of processes, which are generally enumerated and these rarely change. The details might vary over time but the process functions remain the same. For example the claims value stream includes these processes: Initial Reporting, Coverage Verification, Supplier Management, Negotiation and Settlement, Cross-department Claims Billing Settlement, Fraud Management, Litigation Management, and Customer Issue Resolution. These are well known processes with well known functions, the details and data in these processes might change over time, however the functions of these processes remain the same.

The majority of processes and their functions (what each process does end-to-end) are enumerable for an enterprise. In fact a large deviation from these processes and venturing into new areas that are drastically different from the enumerated processes within an enterprise is a major event at an enterprise and is a multi-year plan. The vast majority of enterprise processes (what the process does) are enumerable and remain largely the same.

The prevailing view is if we build or purchase efficient teams and systems, then any process can be built on top of these great teams and systems. Processes are secondary to these functions and systems, as they are considered ever changing, and functions and systems are there to enable any process that the business may envision or desire to build!



“If we have great functions, services and systems, we’re enabling the business to build and operate any process they want!”

Enterprises continue to optimize, improve, and incorporate better functions and systems. Example functions include generating invoices, retrieving claims, approving invoices, making payments, and sending invoices. Example software systems include databases, CRMs, RPA, Workflow tools, cloud services, microservices, data lakes, and others.

The problem is i) processes are well-defined, and not a by-product of the systems used to operate them, ii) efficient teams and systems are not enough to build efficient end-to-end processes.

For enterprise operations the process and its function (end-to-end operations) are equally as important as the individual teams and systems and their individual technology and functions (what they each do).

Each enterprise generally operates a specific set of value chains and processes, in particular in regulated industries, as explicitly stated by their primary activities. An insurance company insures!

For each enterprise most processes are already known and don't change.
For most processes, the majority of the process operations are already known and don't change.

It's time to take a Process first approach in the enterprise!

2.3. Process First Operations

Luther's platform is designed process-first. For efficient enterprise operations, effective end-to-end operations are as important as effective individual services and teams and systems, primarily since the enterprise's core value is delivering a specific set of processes and value streams, particularly in regulated industries, where most value streams & processes are explicitly enumerated!

Enterprise Operations are generally function-first.

They continue to improve functions & systems. Processes are considered secondary.
If we have great functions and systems, the business can operate any process!

Luther's platform is designed process-first.

Primary focus on end-to-end processes.
Reliable end-to-end process operations include consistent operations, & great functions & systems

The most important attributes of process first operations are i) standardized connectivity between all systems involved in the process, ii) Common Operations Script operating the end-to-end process.

Luther's unique value for reliable end-to-end Process Operations is providing

**standard connectivity
a common operations script**

across all teams and software systems.

2.4. Cross-Department Claims Settlement Process in context

Allianz has a number of general Value Streams involved in managing their wealth product lines. One such value stream is "Claims". This value stream includes multiple processes. Cross-department claims settlement is a key process for Allianz, serving to provide insured customers with compensation quickly to maximize customer satisfaction while ensuring timely payment between operational teams to keep operations running smoothly. Its primary purpose is to process the settlement of claims internally, reviewing claims and generating, reviewing and paying invoices between operational entities.

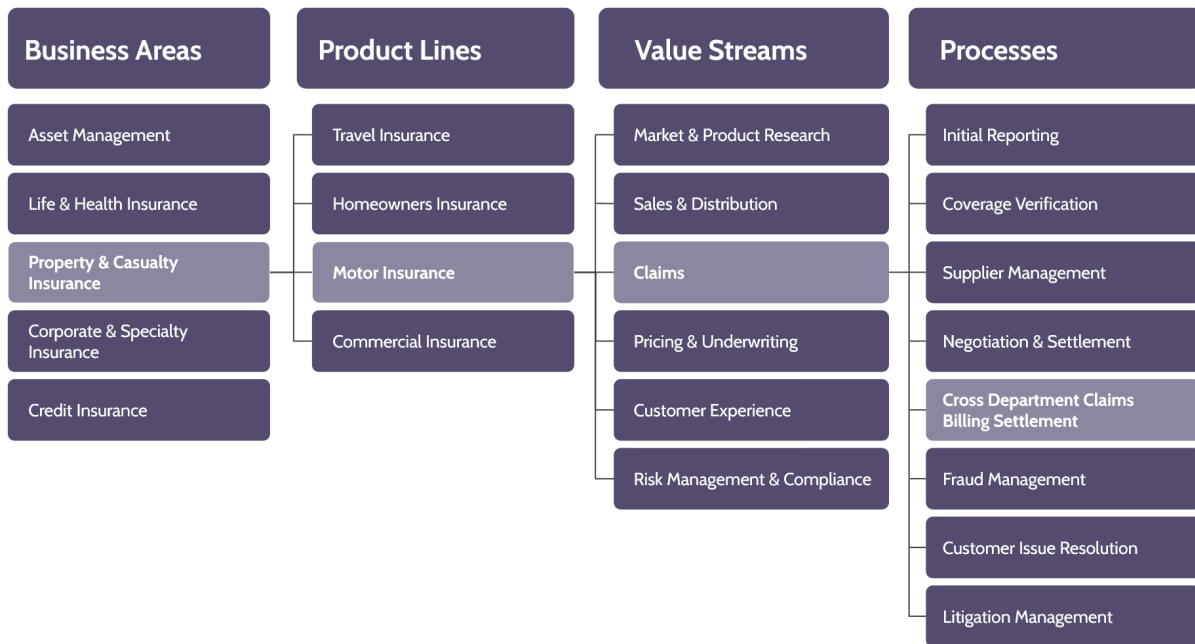


Fig 6. Allianz operates many value streams as part of their operations. Each value stream contains many processes. Claims is a value stream. It contains many processes, including cross-department claims settlement.

Allianz’s cross-department claims settlement process is vital for providing motor insurance. Allianz underwrites motor insurance policies using many OEs. These policies provide coverage across the jurisdiction of different OEs. Often, policies will be issued by one OE and an accident will occur under the jurisdiction of another. Both Allianz teams, one from each OE, are needed to operate the claims settlement process. There are two OEs in this process, the “Handling” OE which operates where the accident occurred, and the “Originating” OE where the policy was issued. The Handling OE handles the claim then issues an invoice to the Originating OE to cover the cost of handling (settlement, supplier costs, document and administrative costs). This creates intra-company debt, which is vital to clear effectively in a timely manner to ensure efficient operations for both entities.

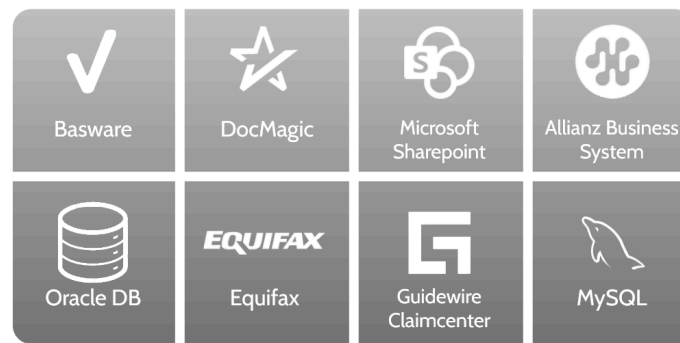
An effective cross-department claims settlement process is essential for paying out to customers in a timely manner, an erroneous settlement process will create errors and delays which will need cost and time-intensive reconciliation and could result in the loss of policy-holders as they seek out other insurance providers. Motor insurance is a key area of operations for Allianz, with 50,000 cross-department claims every year which have a total value of over €200 million.

Annual cross-department claims 50,000	Value of all claims annually €200,000,000	Average value of each claim €4,000
--	--	---------------------------------------

Here, we illustrate the 5 teams each operating a function for the end-to-end process operations:



Each team has a number of software systems. These systems include:



2.5. Cross-Department Claims Settlement Process before

This is the process for cross-department claims settlement with Allianz. Allianz provides vehicle insurance policies for its EU customers, which provides coverage for any accidents. These policies are sold by an individual OE's operating team which covers the customers for any incidents as they travel under different OEs in the Allianz network. This process occurs between separate operational entities, each with their own sets of teams. Here we describe two operational entities, the "Handling" OE which operates where the accident occurred, and the "Originating" OE where the policy was issued. In the claims settlement process, the Handling entity is responsible for handling the claim since they are better equipped to deal with the local requirements and services, while the Originating entity covers the costs incurred.

1. The Handling Claims Team retrieve the claim from the claims database
2. The Handling Claims Team processes the claim and sends it to the Originating Claims Team for review and approval
3. The Originating Claims Team reviews the invoice
4. The Originating Claims Team approves the invoice
5. The Handling Finance Team generate an invoice for the OE, to cover the costs that were incurred by the handling entity while processing the claim, along with a fee for the handling
6. The Originating Finance Team review and approves the invoice
7. The Originating Finance Team sends the invoice to the Originating Payments Team to make the payment
8. The Originating Payments Team makes the payment and completes the claims settlement process.



Fig 7. Illustrates the process of cross-department claims settlement and the participants and systems involved.



3. Problem

3.1. Enterprise Process Operations Problems

Enterprises are complex organizations operating many processes. Enterprises operate processes across fragmented and siloed teams and software systems. This means that teams change their operations as functions and information change, but other teams operating the process are not made aware of that change. Consequently, other teams are operating on constantly changing and incorrect information, resulting in disjointed, inconsistent, inefficient end-to-end operations which lead to high costs, delays and errors. As a result, disjointed process operations require monitoring and reconciliation to correct errors, and this also increases operating costs.

Operational Silos cause unreliable Process Operations

Specifically, operating processes across fragmented and siloed teams and software systems affect process operations both i) technically during the development phase and ii) operationally once they go live in production.

On the technical side, for process changes, enterprises set up case-by-case projects, which includes large development and DevOps teams, and setup of non-standard case-by-case infrastructure and development environments, as well as bespoke connectors between different systems. Further, as the teams and systems change over time they deploy local updates which usually impact the end-to-end operations, requiring further updates and patching.

Technical hurdles in development

- Inconsistent developer process
- Dev teams focus on setup and maintenance
- Large Dev & DevOps teams
- Inconsistent and nonstandard process connectors
- Inconsistent process Ops scripts across teams
- Non-standard infra., connectors, dev. env. setup
- Inconsistent updates & patches over time

Operational problems in production (live)

- Nonstandard ops: across process steps
- Nonstandard ops: over time
- Inconsistent changes
- Lack of execution status visibility
- Need for execution reconciliation
- Large Ops teams
- Compliance fees & violations

Once the process is live, the fragmented and separated teams and systems result in non-standard operations across the process and over time as the teams, operations, and systems change. The fragmentation also results in a lack of execution visibility and operations monitoring. This further results in the execution requiring reconciliation, which is often lengthy and expensive. This could also result in compliance issues and violations. All of this requires large operations teams to run the processes and fix their recurring issues.

3.2. Problem Overview

For reliable process operations, all teams and systems involved should operate the same end-to-end Process.
They often don't!

Allianz's cross-department claims settlement process operates across teams in different operating entities (OEs) that have poor visibility of operations, due to varying data standards and bespoke documents across OEs, as well as non-standardized data entry. Due to siloed operational teams, employees spend a considerable amount of time manually correcting data errors on these documents and performing reconciliation with other teams, which causes delays and increases operational costs for the process.

3.3. Cross-Department Claims Settlement Process Operations Problems

Allianz's cross-department claims settlement process operates across many OEs, each with their own operational teams. Each of these OEs is operationally separate, siloed and separate from each other. As a result, there is poor visibility of operations between teams, which leads to several problems.

Different OEs have different data regulations, formats, and standards. This means that claims documents and invoices will be different across jurisdiction boundaries of OEs. As a result, Allianz requires employees to manually check these documents and reconcile errors, which extends the process' operating time, causing delays. Additionally, manually checking these documents and changing them to match the correct OE's formats introduces a source of errors, which need to be corrected later in the process, further increasing delays and increasing operating costs.

When handling cross-department claims, the Handling entity incurs costs associated with operating the claim, and they invoice the Originating entity, who is responsible for covering the invoice. This means that for the period between handling the claim and the invoice being paid, the Handling entity is operating at a loss. With efficient operations, this should not be a problem as the Handling entity will be paid in a timely manner. However, inefficient operations and siloed teams mean delays of weeks or even months can occur in payment. If the Handling entity is handling many cross-department claims, they can quickly accrue large amounts of operating debt, which will prevent them from operating further, causing further delay to customers, delaying claims further. These delays to the process prevent customers from being recompensed in a timely manner, which lowers customer satisfaction and trust in Allianz, damaging their reputation. This is a big problem for Allianz, who are the largest insurer in the world.

Additionally, siloed teams and poor visibility of operations means that the documentation for each claim and invoice is often scattered across multiple teams and OEs, in multiple formats. Additionally, due to manual reconciliation, the documents may be erroneous or incomplete. As a result, this makes carrying out audits on claims very difficult, which makes it harder for Allianz to comply with financial regulations, which may lead to penalties.

Practical Problems in the settlement process

Settlement delays lower customer satisfaction

Differing policy processing leads to inconsistent service

Lack of transparency leads to customer confusion

High outstanding debts impede operations

Difficulties in performing audits efficiently

Damage to Allianz's reputation across Europe

Inconsistent fraud detection

Operational Problems for Allianz

Nonstandard ops: across process steps

Nonstandard ops: over time

Inconsistent changes

Lack of execution status visibility

Need for execution reconciliation

Large Ops teams

Compliance fees & violations



4. Traditional approaches to process operations and automation solutions don't work

4.1. Approach to Process Operations today

Enterprises typically establish dedicated projects and project teams to set up process operations. This involves mobilizing large development and DevOps teams, as well as large operations and support teams. They create custom, often non-standard project infrastructure, connectors, and development environments, which require dedicated ongoing maintenance once the process is live. The project team writes bespoke operations code to manage the end-to-end process, including code that links the operations of various software systems.

As the process moves into production, developers must continuously write custom local code to adapt to the evolving landscape of team operations, process rules, and software systems. Additionally, the project team or other development teams need to develop and integrate separate execution monitoring software and reconciliation software. These tools are essential for detecting errors and inconsistencies, determining root causes, and correcting the issues. Furthermore, they deploy multiple distinct application systems, such as compliance software systems, to support the overall operation.

This demonstrates the bespoke, fragmented nature of process operations development, in addition to multiple auxiliary systems required to keep the operations going. Most importantly, this approach cannot keep pace with the ever-changing process operations.

As a result, Enterprise process operations are unreliable!

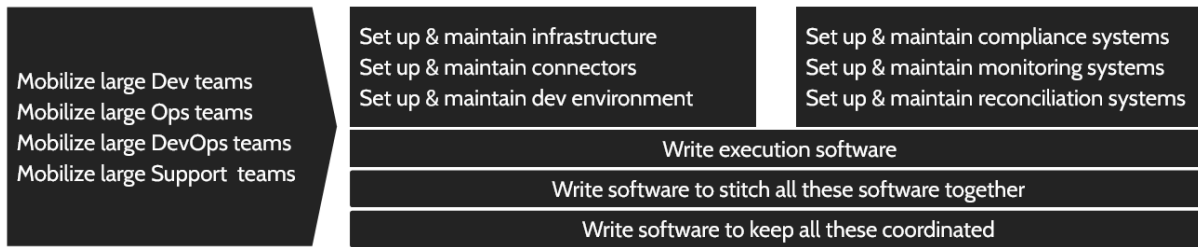


Fig 9. Enterprises generally carry out all of the above to run a process.

4.2. Bespoke Connectors & Operations Scripts & why they don't work

To manage the cross-department claims settlement process, Allianz typically; i) sets up local connectors between directly linked systems involved in the process, ii) Develops and updates local operations scripts to manage the process end-to-end. Both the bespoke connectors and operations scripts require regular updates and modifications as teams, process operations, and software systems evolve. These updates are reactive and localized, addressing immediate changes without fully considering the entire process.

The problem arises because these connectors and scripts are integral to the end-to-end process, where each step depends on others and assumes specific functions from other parts. Local changes alter the immediate local operations, but the rest of the process continues to rely on outdated assumptions about those functions. This results in a gradual drift and fragmentation between different parts of the process.

This drift and fragmentation requires further patches and updates, which will require further patches and updates in other parts of the process, and the cycle continues!

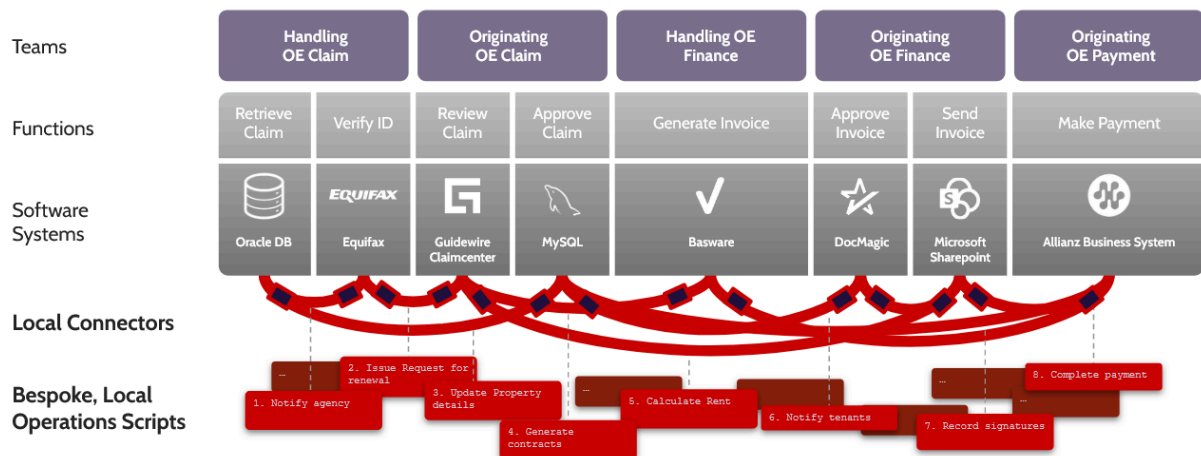


Fig 10. Bespoke local connections across the end-to-end process that are internally developed by the enterprise.

4.3. Local Automation (RPA, Workflow) tools & why stitching them together doesn't work

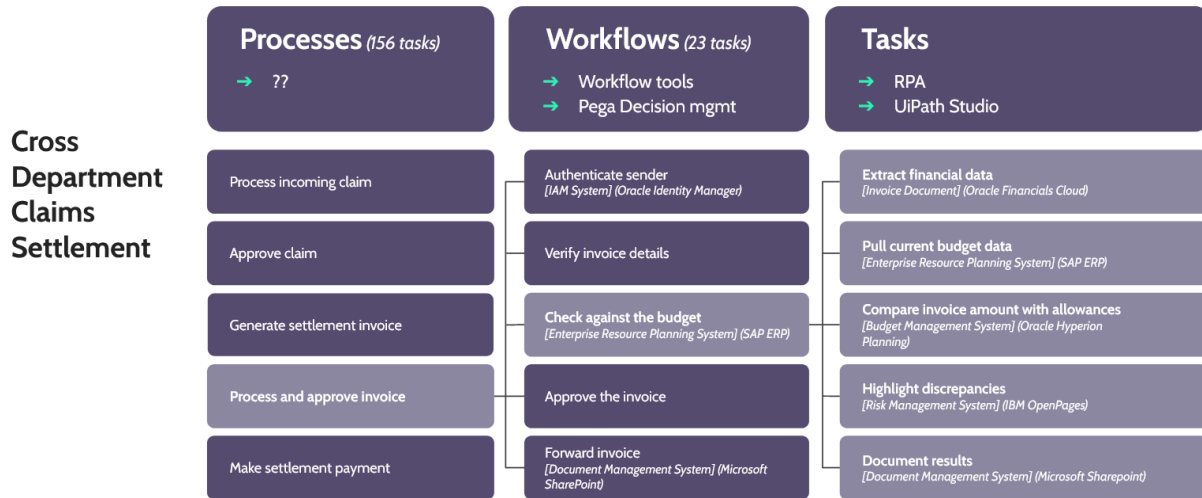


Fig 11. Today, there are no traditional tools which effectively automate processes.

Enterprise processes consist of numerous operations (tasks). Each process includes a collection of workflows, and each workflow is a collection of multiple tasks. Tasks are simple, localized events involving one team and one or two software systems. For example, inputting data into an invoice is a task. Workflows are more complex, comprising 10–20 tasks that span one to two teams and involve two to three software systems. For instance, processing a claim to send to another team is a workflow consisting of 23 tasks. Processes are complex, involving over 50 tasks, three or more teams, and multiple software systems. An example of a process is the settlement of a cross-department claim, which includes 156 tasks.

Enterprises utilize Robotic Process Automation (RPA) tools to automate individual tasks. RPA tools have evolved into highly effective solutions for this purpose. However, for automating workflows (comprising 10–20 tasks), enterprises turn to Workflow Automation tools, as individual RPA bots are not scalable to handle such complexity. Workflow Automation tools have similarly advanced, becoming highly effective at automating entire workflows. These tools leverage a diverse array of technologies, including traditional ones like Workflow tools, ERPs, and BPMs, as well as modern innovations such as Hyper Automation, Intelligent Automation, and various developer tools.

RPA tools and Workflow tools do not scale to operate end-to-end processes

To overcome the limitations of the traditional approach, enterprises deploy numerous RPA and Workflow tools across the end-to-end process, and then connect and orchestrate these tools to function reliably. This integration and coordination are typically developed internally by the enterprise.

Process orchestration approaches integrate combinations of RPA and workflow systems using point-to-point message passing techniques. These services often employ a batch scheduler or workflow system, which effectively coordinates tasks within a single team. However, this method falls short for processes involving multiple teams. Each team tends to create bespoke code for their tasks, leading to “script bloat” – the proliferation of numerous, often redundant, and poorly documented scripts. This complicates maintenance and scalability. Furthermore, there is a lack of transparency between participants in the process. This lack of coordination and integration results in inefficiencies and errors, causing delays and operational friction. For a full explanation of traditional process operations and Luther’s solution, request access to the [Deep Process Automation Primer](#).

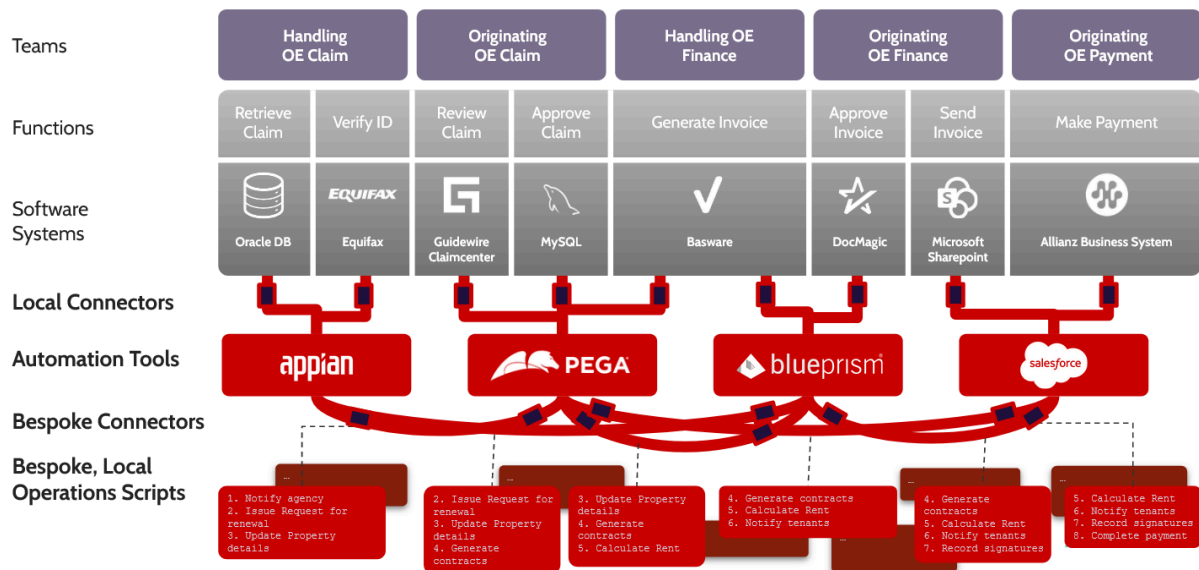


Fig 12. Stitching together local automation tools through local RPA and workflow tools is messy, localized and ultimately unreliable.



5. Solution

5.1. Luther's Platform

Luther's Platform was used by Allianz's development team to build an exemplary cross-department claims settlement system that allows Allianz to avoid mismatched data formats, reduce data errors, whilst providing a single source of truth for claims that reduces errors. This required a system that could effectively handle the operations of multiple teams, software systems, numerous tasks and validations end-to-end, as well as taking into account different regulations and data standards across the jurisdiction of different OEs.

Luther's unique value for reliable end-to-end Process Operations is providing

**standard connectivity
a common operations script**

across all teams and software systems.

This is very difficult and costly with traditional automation tools and workflows. Automation of the cross-department claims settlement process requires Luther's Deep Automation Platform. The automated cross-department claims settlement process is the result of this work and is an end-to-end claims settlement system that standardizes the process, while reducing settlement operating costs and timescales, reducing the need for manual intervention to check for errors, and improving the customer experience by reducing claims times.

5.2. How it works on the Luther Platform

The Luther Platform standardizes data formats and data entry across OEs for Allianz claims. The platform can automatically verify and approve invoices based on predetermined rules stored on the platform, which saves time for manual reconciliation. Importantly, all teams and systems still perform their function. One key feature of the Luther Platform is that teams operating the process can update their operations scripts as before, and the platform doesn't complicate or change the way the process is operated. With the Luther Platform, when a team does decide to change their operations, all other teams are automatically notified. The Luther Platform simply streamlines manual operations and improves cross-team visibility, turning siloed operations into a cohesive end-to-end process.

How it works on the Luther Platform:

1. The Handling Claims Team retrieve the claim from the claims database
2. The Handling Claims Team processes the claim and sends it to the Originating Claims Team for review and approval
3. The Originating Claims Team reviews the invoice automatically using rules stored on the Platform
4. The Originating Claims Team approves the invoice automatically using rules stored on the Platform
5. The Handling Finance Team generates an invoice automatically using the Platform for the OE, to cover costs that were incurred by the handling entity, along with a fee for the handling
6. The Originating Finance Team review and approves the invoice automatically using rules stored on the Platform
7. The Originating Finance Team sends the invoice to the Originating Payments Team to make the payment
8. The Originating Payments Team makes the payment and completes the settlement process.

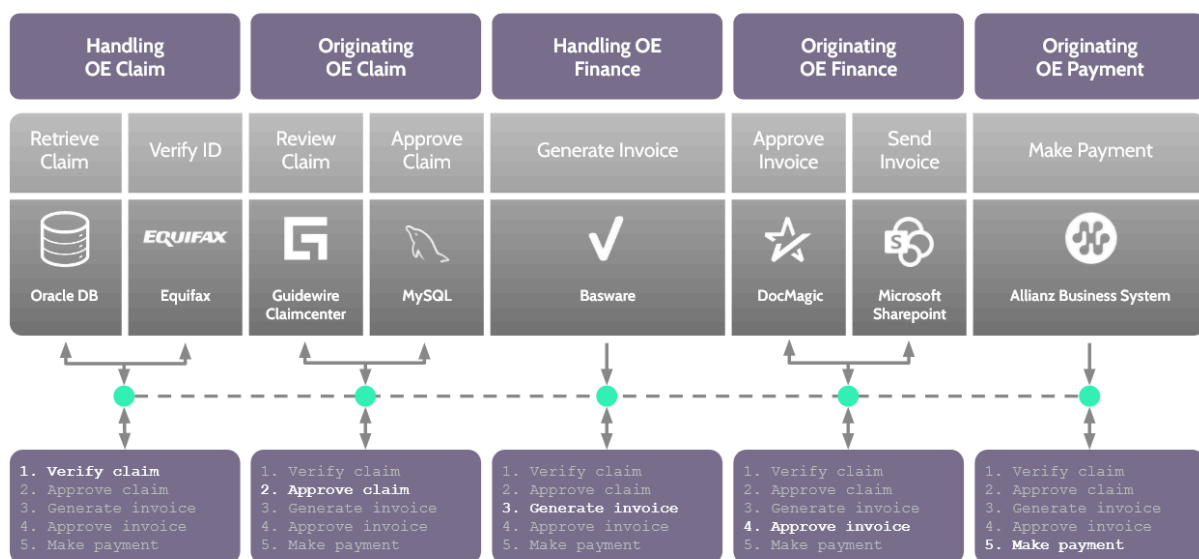


Fig 13. Overview of the Luther Platform automating the cross-department claims settlement process.

For a more detailed view of the steps operating the Luther Platform, please view the appendix.



6. Implementation

Luther's team worked with the Allianz team to implement the automated cross-department claims settlement Process on the platform.

First, Luther's team worked with the teams at Allianz to map the process. View a process map [here](#). Luther then identified all teams and all software systems involved in the operations of the process. Luther then allocated a node to each team, deployed the platform on all nodes, and connected the nodes to each of the software systems, through Luther's standard connectors. Then Luther's team worked with Allianz developers to develop a robust common operations script for process operations. Then the application went live.

For more information please visit these links, for [implementation steps](#), [implementation in general](#), and [sandbox](#).

Customer Team	<i>Business Owner, Application Owner, Technical Lead</i>		<i>Day 1</i>
Discover	<i>Phase 1</i>	<i>Describe process operations</i>	<i>2 weeks</i>
	<i>Phase 2</i>	<i>Describe systems & technical requirements</i>	
Process mapping	<i>Map the process</i>		<i>2 weeks</i>
Platform set-up	<i>One-time platform set-up</i>		<i>1 day</i>
Build application	<i>Develop (code) application operations</i>		<i>8 weeks</i>

Fig 14. Implementation timeline for the automated Cross-Department Claims Settlement Process.

To implement the cross-department claims settlement process, Luther and Allianz followed these steps:

6.1. Process mapping

Luther's team worked with multiple Allianz teams to map the process operations. The process map includes i) functions, ii) data inputs and outputs at each step, and iii) rules and decisions at each step. Teams are operationally separate entities involved in the process. As part of process mapping, Luther identified the exact set of software systems and teams involved in operating the end-to-end process.

6.2. Identify teams and software systems

Luther's team identified the teams and participants involved in end-to-end process operations. The teams are: the Handling Claims Team, the Originating Claims Team, the Handling Finance Team, the Originating Finance Team, and the Originating Payments Team.



Fig 15. Luther's team worked with Allianz to map the process including 6 teams involved in end-to-end operations.

Luther's team identified the software systems involved in end-to-end process operations. These systems are: Oracle Database, Equifax, Guidewire Claimcenter, MySQL, Basware, Microsoft Sharepoint, DocMagic, and Allianz Business System.

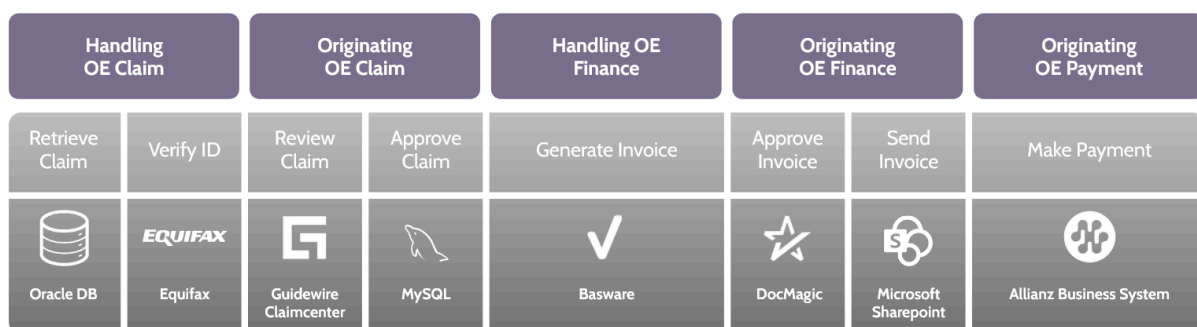


Fig 16. Luther's team identified the software systems involved in the end-to-end process operations.

6.3. Nodes and Connectivity through distributed system for end-to-end team connectivity

Luther's team assigned a dedicated node to each team involved in the process by allocating servers to their respective teams. These servers are cloud-native and can be deployed on either public or private clouds, depending on security requirements. All nodes are interconnected through a distributed system, which facilitates the sharing and validation of operational functions and data among all teams.

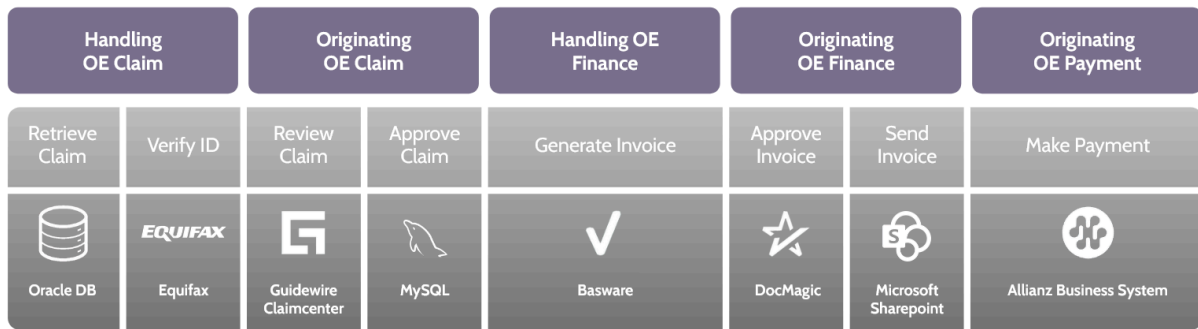


Fig 17. Nodes are connected via a distributed system on the Luther Platform.

6.4. Connectors to software systems

Each team has a number of software systems involved in its operations, as identified in the process map. For each team, Luther's platform connects its node to all software systems involved in its operations. Luther has a set of standard connectors across a wide range of enterprise systems, which the Luther platform deploys to rapidly connect to the systems involved in operating the process. This is done by determining the technology, type and system of the connector to connect to each system in the process.

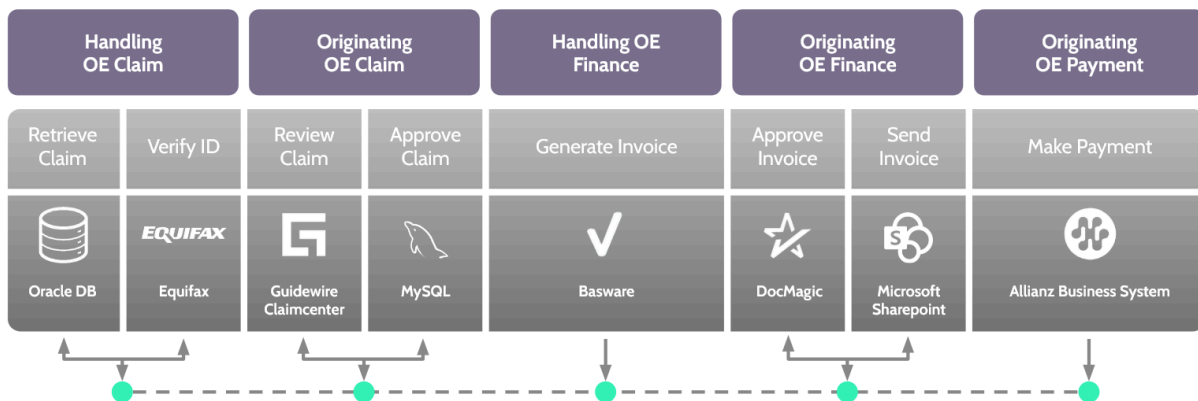


Fig 18. Luther's team set up connectors that link the processes together.

Luther, through numerous enterprise implementations has standard connectors to a majority of enterprise software systems across a range of processes and industries. For a full list of our connectors, please visit: ["Luther Platform Connectors"](#).

6.5. Platform set-up

The Allianz team selected a set of configurations for their platform specifications. This selection depends on i) the process complexity (number of tasks), ii) amount of data processed (KB) per process run, iii) number of participants, iv) reliability, availability and security requirements for the application. Based on these selections, Luther's team deployed the platform on all nodes. For more details on platform configuration specs please visit: ["Luther Platform Connectors"](#).

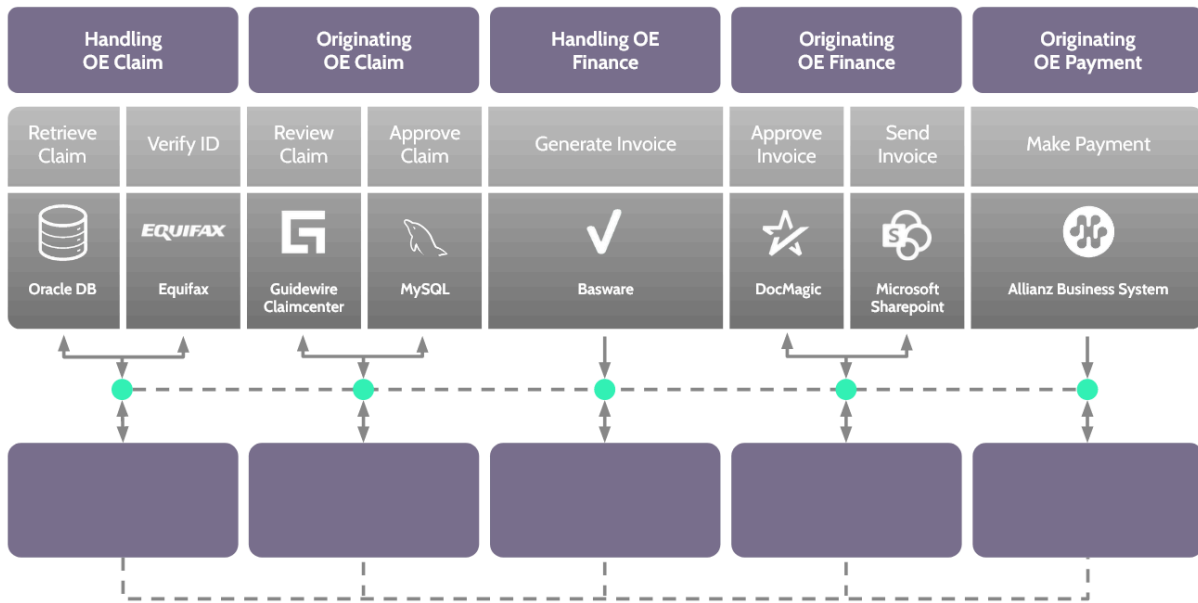


Fig 19. The platform is set up on each of the nodes, ready to reliably operate the end-to-end process at each step.

Luther's platform vertically integrates distributed system technology, optimal resource allocation and management, real-time event ordering and streaming (sharing), and deterministic event processing and execution, to provide a modern technology stack to reliably operate an end-to-end process across multiple software systems, at scale.

6.6. Common Operations Script for process operations

The platform is now fully set up and connected with all systems involved in the operation. The Allianz development team, in collaboration with Luther, developed the Common Operations Script to manage the end-to-end process. Connectors translate data from local systems into a common data model utilized by the Common Operations Script. This script encapsulates the business logic, data, rules, and validations for each process step.

Luther's platform vertically integrates

- distributed system technology**
- optimal resource management**
- real time event streaming**
- deterministic event execution**

To make reliable end-to-end process operations possible.

The Common Operations Script effectively codes and operates the process map, executing the Operating Cycle for each system across the entire process. To operate the process end-to-end, each function performs the same cycle of steps: i) send data & information to the System, ii) receive response from the System, iii) compute & validate response, iv) share & store execution of step, v) evaluate & initiate next steps.

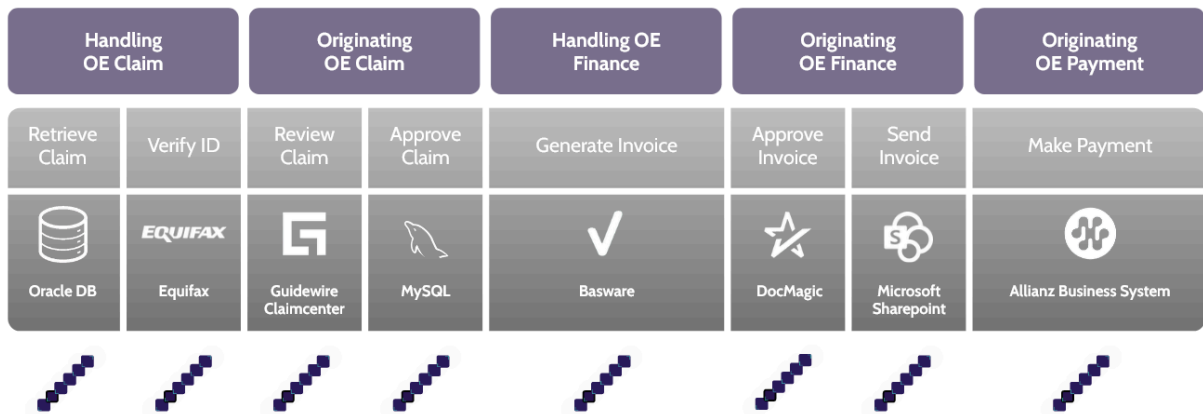


Fig 20. These requirements repeat for all functions across the end-to-end Process Operations.

For a more detailed description of how the Common Operations Script operates the Process please see the Appendix.

This script is shared by all participants and operates on the Luther Platform. Each participant can change the script through suggesting changes, once the changes to the script are approved by all participants the script is updated for all participants.

Teams & Systems can easily change the Common Operations Script

They propose changes
All other teams know and agree to the change upfront
The Common Operations Script is updated for everyone

The enterprise has full autonomy over the process operations to modify and change them, and it also ensures all participants are operating “the same process” at all times. When a team changes their operations, the operations for all participants are updated simultaneously. For a demo of the build process please visit our [website](#).

All teams & systems involved operate the same end-to-end process all the time!
The enterprise has full autonomy over its Operations & Operational changes

So, consistent changes are not an afterthought in a memo
No need to call someone everytime you want to make a small change!

6.7. Go live (production)

Once the platform is set up and the Common Operations Script is coded, the application is ready to go live. Upon going live, it automates the operations of the end-to-end cross-department claims settlement process by providing i) standardized connectivity between teams and systems, ii) the Common Operations Script, shared by all teams, ensures a consistent process operation at all times. For more information about Luther's platform please view this [video](#).

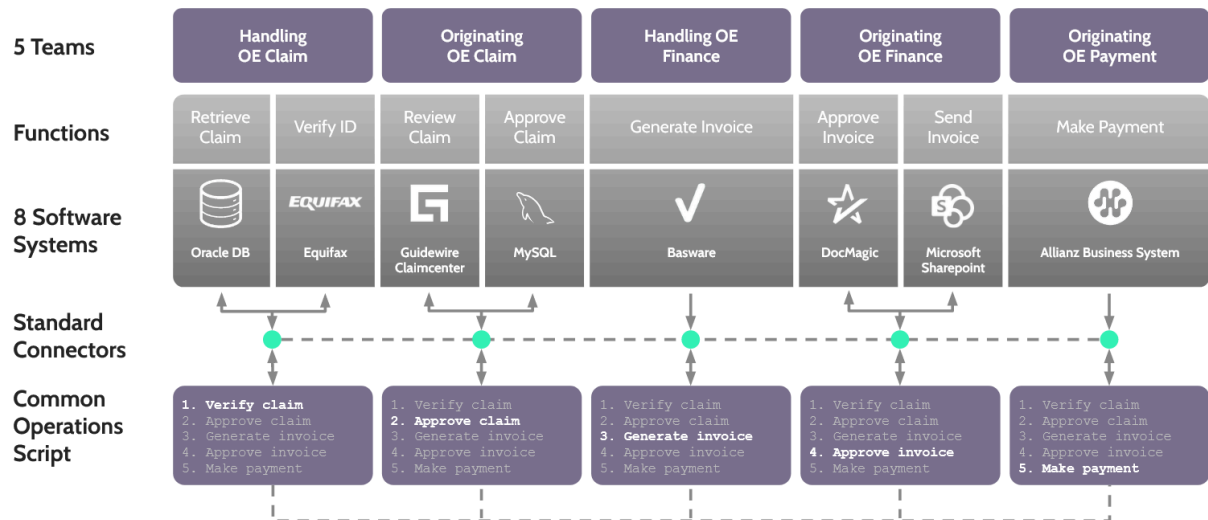


Fig 21. Luther and Allianz developer teams work together to write the common operations script, converting siloed tasks into a shared, transparent workflow that links every step in the process. The common operations script links independent systems into one cohesive process.



7. Results

7.1. Commercial results

ROI: 14X

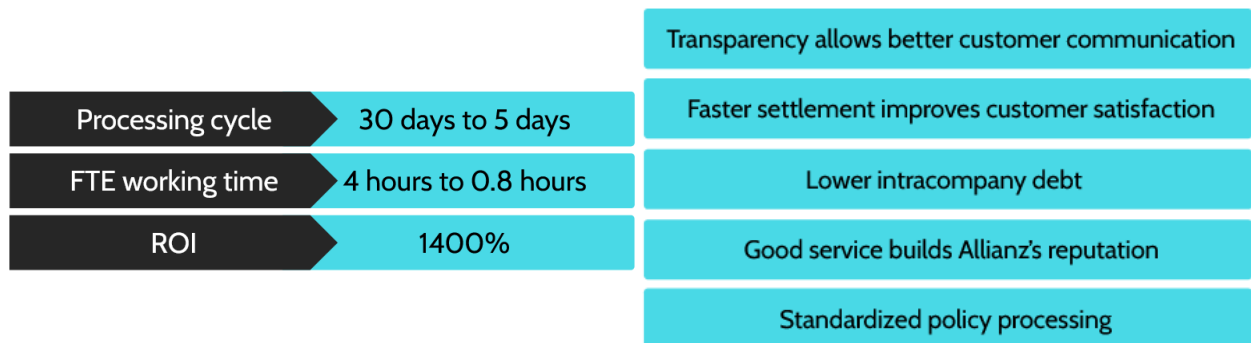
Cost Reduction: 80%

Average Processing Speed Increase: 5X

Using Luther's Deep Process Automation platform, implementing the automated cross-department claims settlement process reduced the cost of the cross-department claims settlement process by 80%. This is primarily due to FTE savings in both ops teams involved in process operations as well ops teams involved in correcting data errors and communicating with other Allianz teams to solve data issues. Automating operations reduced errors associated with the process. The average total working time for the cross-department claims settlement process was reduced from 4 hours to just 0.8 hours, speeding up the average processing time by 5X. This results in a return on investment of roughly 1400%. Luther's solution for cross-department claims settlement doesn't just connect individual software systems and teams. It provides a platform that is flexible and scalable to the changing process for years to come, facilitating process operations where teams can focus on the tasks, rather than reconciling with other teams and causing large delays with new software systems.

Specific commercial advantages:

- Average total time for claims settlement reduced from nearly 30 days of delays to just under 5 days
- Average total working time for FTEs reduced from 4 hours per claim to just 0.8 hours
- fast processing times for claims, leading to better customer relations
- Cost savings of 80%, reducing the operational costs by millions of euros a year, resulting in an ROI of 1400%



7.2. Operational benefits

Luther delivered a platform that standardizes claims settlement, while reducing inefficiencies, improving process transparency, reducing the size of operations teams, and improving compliance, which could not have been achieved without Luther's Deep Process Automation Technology.

General operational advantages

The Luther Platform streamlines operations across enterprise processes, reducing process time and cost while maintaining transparency and flexibility.

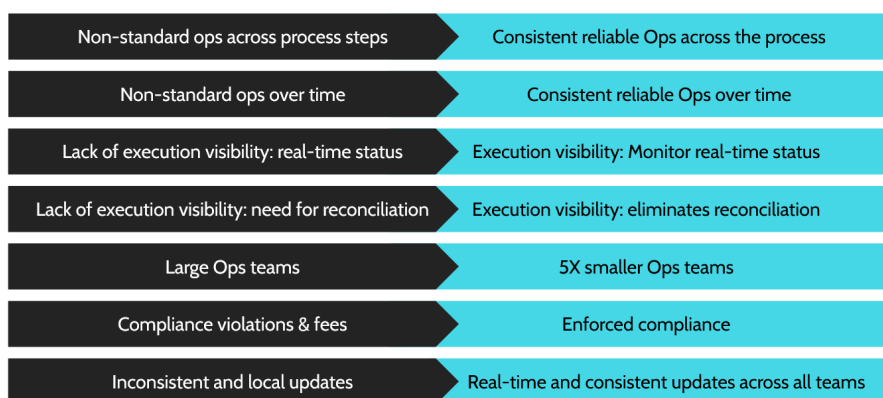


Fig 22. General results from implementation of the Luther platform

Specific operational advantages

Implementing the automated cross-department claims settlement process has streamlined process operations, making it more efficient, faster, and standardized all while requiring minimal manual reconciliation. The platform is flexible and scalable to future changes to the process or data or document formats.

Enhanced cross-department claims settlement operations:

- Increased process reliability and fewer processing errors eliminate costs associated with improperly filed claims
- Elimination of manual reconciliation means smaller operational claims teams at Allianz
- Standardization of the claims process, ensures faster timescales improving customer satisfaction and reducing operational costs for claims teams

7.3. Technical benefits

General technical advantages

The Luther Platform makes process operations more consistent as well as standardizing the infrastructure used to operate the cross-department claims settlement process. Real-time updates across the end-to-end process ensure less downtime in the process, improving efficiency. All this means that developer and developer operations teams can be reduced in size and that developers can focus on developing and improving process operations rather than focusing on handling inefficiencies in the process.

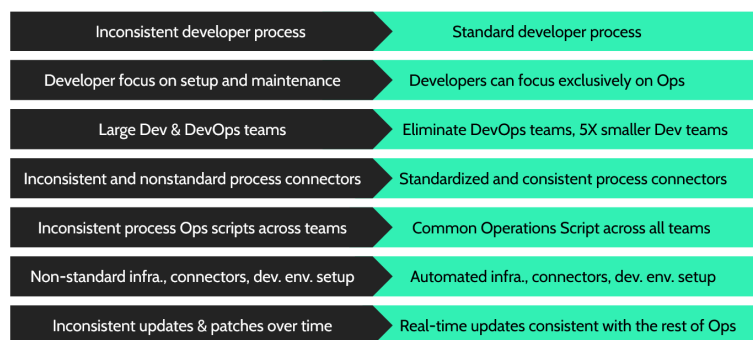


Fig 23. General technical results from implementation of the Luther platform.

Specific technical advantages

Improved operating efficiency:

- Automatically verifies execution to increase reliability and reduces processing errors
- Common execution visibility to all participants reduces troubleshooting effort
- Automatically supports Common Operations Script or Platform updates including new data regulations

Improved compliance and data storage:

- Single source of truth for data across organizational boundaries
- Full user control over access to data and data sharing



8. Expansion

The ICS product built on the Luther Platform effectively standardizes the cross-department claims process. The Luther Platform provides a stable foundation for building new applications for Allianz operations and could be used in several ways to streamline and automate Allianz operations. Further expanding the product to include other aspects of the claims and insurance pipeline could expand Allianz's market presence, allowing them to take control as a market leader in other aspects of the insurance value stream.

Potential areas of expansion to further automate Allianz's operations of the Luther platform include:

- Use the Luther Platform to offer customers a streamlined car purchase experience that includes the purchase of insurance as one seamless process
- Processing accidents from first notice of loss, expanding the current claims process
- Processing communications with suppliers and third-party repair centers, extending the product to include more aspects of the claims processing life cycle



9. Luther Company & Offerings

9.1. What Luther does

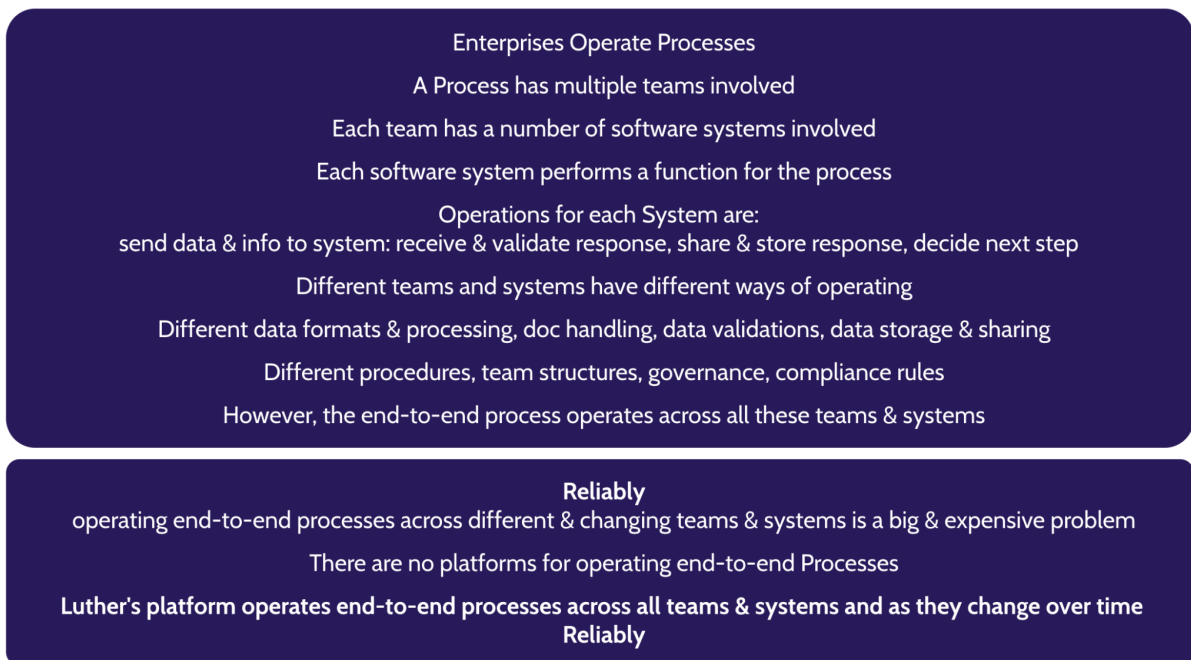


Fig 24. Luther's platform solves the complicated problem of end-to-end enterprise process operations.

For more information about Luther, please visit our [website](#).

9.2. “In a nutshell” - Luther’s unique value

Luther’s platform vertically integrates

- distributed system technology
- optimal resource allocation and management
- real time event ordering and streaming (sharing)
- deterministic event processing and execution

To make reliable end-to-end process operations possible.

Luther’s unique value for reliable end-to-end Process Operations is providing

- standard connectivity
- a common operations script

across all teams and software systems.

Luther automates the operations of the end-to-end process across multiple teams and systems

9.3. Platform implementation

To implement the Luther Platform, organizations work with Luther through an implementation process - laying out objectives and expectations for the project, then mapping the process and setting up infrastructure. After this, enterprise developers build code that will execute the agreed process.

Customer Team		<i>Business Owner, Application Owner, Technical Lead</i>	<i>Day 1</i>
Discover	<i>Phase 1</i>	<i>Describe process operations</i>	<i>2-4 weeks</i>
	<i>Phase 2</i>	<i>Describe systems & technical requirements</i>	
Process mapping		<i>Map the process</i>	<i>1 week</i>
Platform set-up		<i>One-time platform set-up</i>	<i>1 day</i>
Build application		<i>Develop (code) application operations</i>	<i>4-8 weeks</i>

Fig 25. Implementation timeline for an application operated on the Luther Platform..

Enterprises working with Luther fill in the details of all software systems and connectors for their processes. These documents are used to build the process map and subsequently, the application.

Item	Software System	Type	Category	Connector Technology
System 1	Oracle Database	Both	ETL	Oracle DB 21c
System 2	Guidewire Claimcenter	Both	Industry Specific	Guidewire Integration Framework
System 3	Equifax	Both	Industry Specific	Equifax REST API
System 4	MySQL	Sink	ETL	MySQL Connector
System 5	Basware	Sink	Payments	Basware Network REST API
System 6	Microsoft Sharepoint	Sink	ETL	SharePoint REST API
System 7	DocMagic	Sink	Agreements	DocMagic API
System 8	Allianz Business System	Both	Industry Specific	CISL (Core Insurance Service Layer) REST API

Fig 26. The list of software systems involved in end-to-end Cross-Department Claims Settlement Process operations

Build Distributed Ledger				
Item	Detail	Description	Input	Comments
Network	Number of organizations	These are separate IT teams that may be internal or external to one another.	5	Each participant belongs to a separate organisation.
Network	Number of peers per organisation	This determines the reliability of executing the process.	2	Each participant runs 2 peers for high availability.
Network	Number of peer cores	This is determined by the complexity of the process.	4	Each worker has 4 cores to process 10 claims per second max throughput.
Orderer	Number of Orderers	Number of orderer service instances.	3	Spread orderers across 3 availability zones for high availability and practically 100% system uptime
Orderer	Number of orderer cores	Number of cores allotted for each orderer instance.	2	Allow enough cores to support 10 claims per second max throughput.
Resource Management				
Virtual Machines				
Item	Description		Input	Comments
Number of Cores per Instance	Number of cores per instance in the cluster worker pool.		4	Ensure each peer has 2 cores for parallel event processing.
Ledger Size (GB)	Size of volumes used to store the ledger.		100	Provide enough storage for a years worth of transactions without resizing
Number of Worker Instances	Number of worker instances to utilize in the cloud region, distributed across availability zones.		5	One worker per participant
Cloud				
Item	Description	Specifications	Comments	
Cloud Provider Name	Cloud Service Provider that the platform is deployed into.	AWS	Deploy on AWS.	
Cloud Service Account	What cloud service account will be used for deployment?	141812438321	Use existing AWS account.	
AWS Role ID	Only necessary for AWS.	arn:aws:sts::343039485463:role/admin	Use role that requires MFA for InfoSec requirements	
Cloud Provider Region	A cloud-specific string identifier for a geographic region.	us-east-2	Closest to customers	
Cloud Provider Domain	A string identifier for a company domain	ford.luthersystemsapp.com		

Fig 27. A sample list of connectors and infrastructure, similar to one an enterprise building an application on the Luther Platform would fill out.

9.4. Results of the Luther platform for Process Operations Automation

At Luther, we recognize that enterprise processes of today are complex and challenging to automate. We provide a platform for successful process automation.

The results are incredible. Enterprises working with Luther see an average return of 10 times their investment. Time is saved everywhere, with development of process applications and automation technology sped up by 2.5 times, and processing times 7 times faster. Find out more about Luther's core platform features [here](#).

2.5X faster development

10X less operational costs

7X faster processing time

10X ROI

1000s of compliance rules automated

9.5. Luther's platform architecture



For a more detailed introduction on the Luther platform please request access to the "[Luther Deep Process Automation Primer](#)".

For a detailed introduction and documentation examples please see the [Luther Platform site](#).

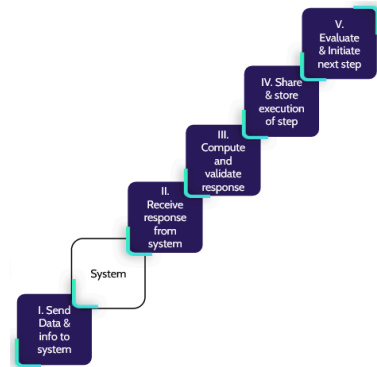
For more information about Luther's platform please visit luthersystems.com.

10. Appendix

10.1 How the platform operates an end-to-end process: Application walkthrough

Below is a more detailed walkthrough of the process operations, across the teams and software systems. Each step in the process follows the exact same 5 operational substeps which the Platform executes:

- i) **Send:** Platform sends data & information to the System,
- ii) **Receive:** Platform receives response from the System,
- iii) **Validate:** Platform computes & validates the response,
- iv) **Store:** Platform shares & stores execution of step,
- v) **Evaluate:** Platform evaluates & initiates next steps.



The Common Operations Script ensures that these operations cycle steps are carried out for all systems involved in the process to ensure reliable process operations.

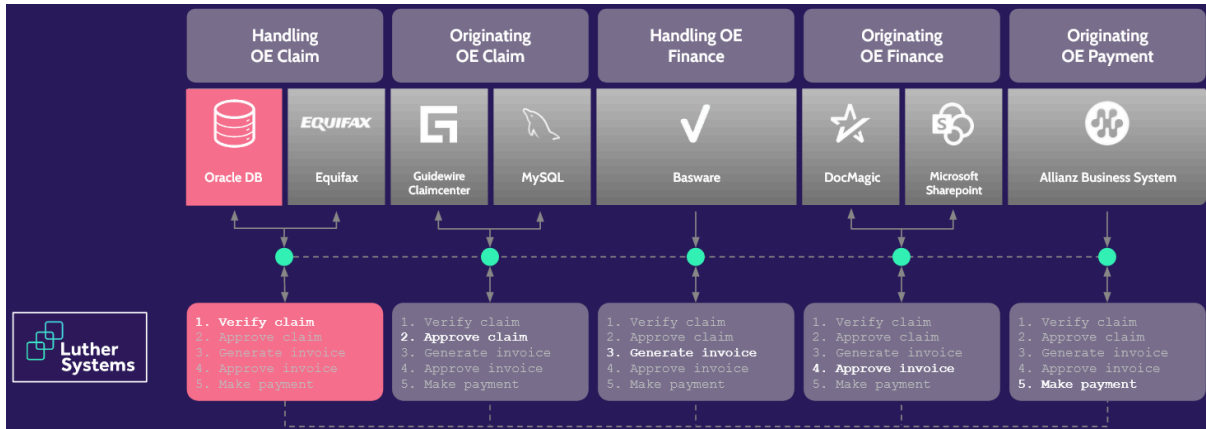
The Platform operates the Process by standardizing the execution of each step in section 5.2. “How it Works on the Luther Platform”

The process involves 8 different software systems, the systems’ functions are as follows:

- Oracle Database: In Step 1 it retrieves the claim from the claims database
- Equifax: In Step 2 it verifies the claimant and claim details
- Guidewire Claimcenter: In Step 3 it reviews the claim
- MySQL: In Step 4 it approves the claim
- Basware: In Step 5 it generates the invoice for the originating entity to pay
- DocMagic: In Step 6 it approves the invoice
- Microsoft Sharepoint: In Step 7 it sends the invoice between Allianz departments
- Allianz Business System: In Step 8 it makes the payment from the originating entity to the handling entity

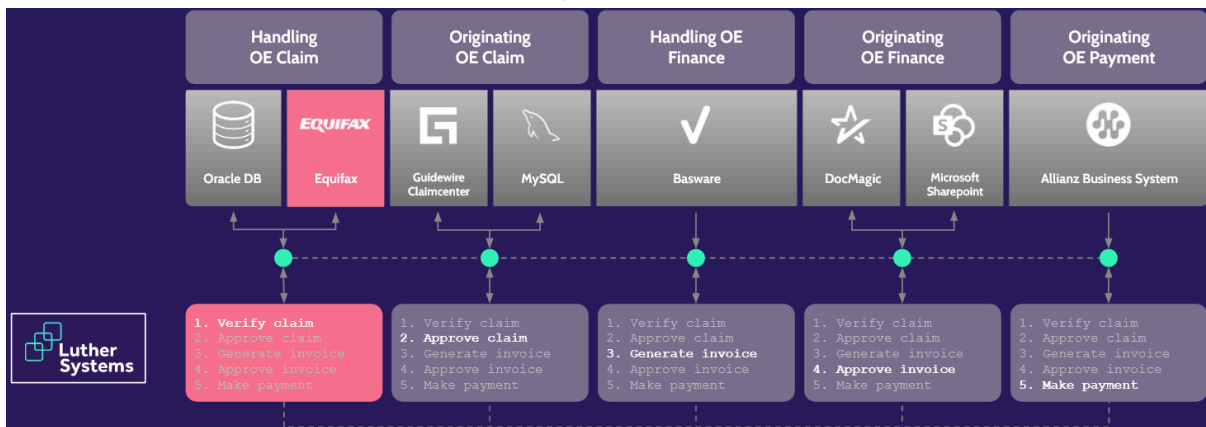
Step 1: Handling Claims team executes Process Claim, specifically Retrieve Claim

- I. Platform sends *claim ID* (request) to Oracle Database
- II. Platform receives *claim data* (response) from Oracle Database
- III. Platform validates *claim data* based on predetermined rules in the Common Operations Script
- IV. Platform shares & stores *claim data* from Oracle Database
- V. Platform evaluates & initiates next step



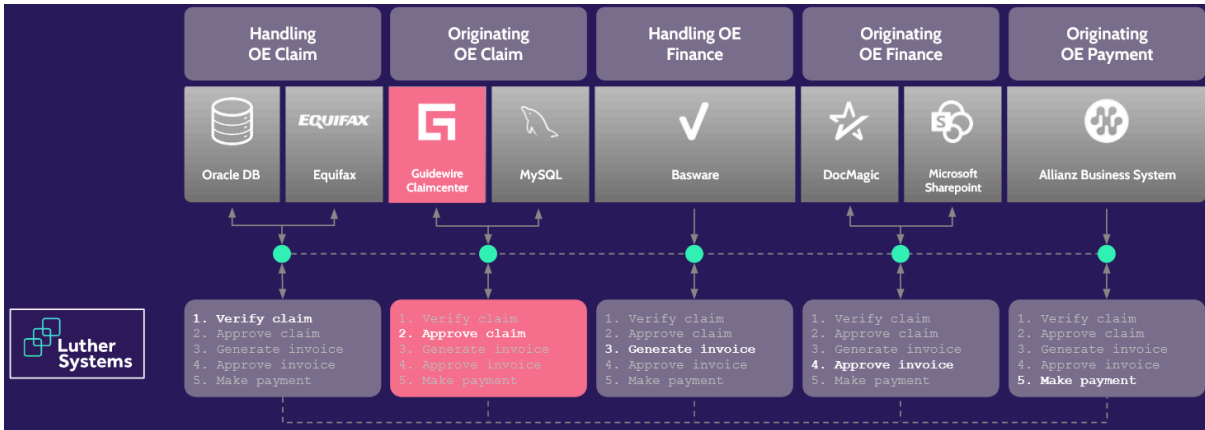
Step 2: Handling Claims team executes Process Claim, specifically Verify ID

- I. Platform sends *verification confirmation* (request) to Equifax
- II. Platform receives *verification confirmation* (response) from Equifax
- III. Platform validates *verification confirmation* based on predetermined rules in the Common Operations Script
- IV. Platform shares & stores *verification confirmation* from Equifax
- V. Platform evaluates & initiates next step



Step 3: Originating Claims team executes Approve Claim, specifically Review Claim

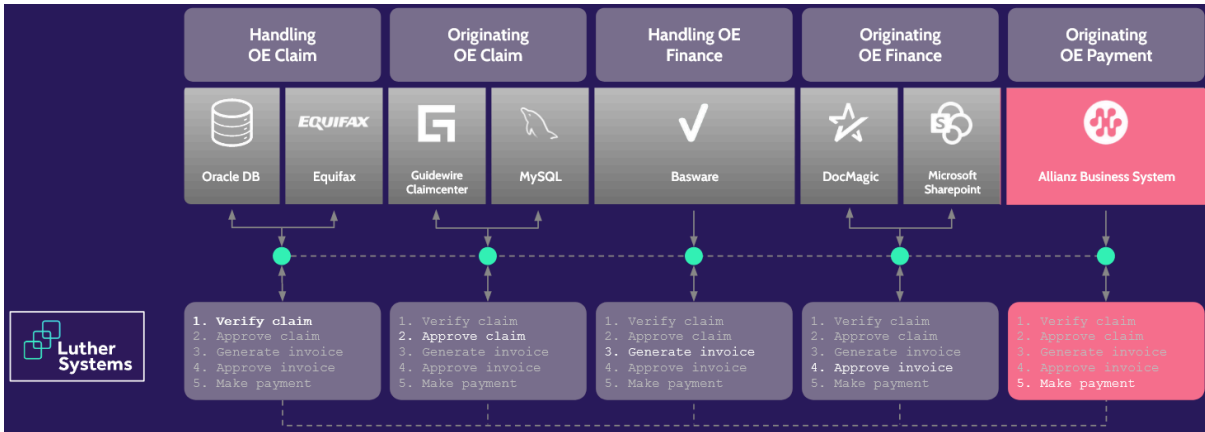
- I. Platform sends *claim review* (request) to Guidewire Claimcenter
- II. Platform receives *claim review* (response) from Guidewire Claimcenter
- III. Platform validates *claim review* based on predetermined rules in the Common Operations Script
- IV. Platform shares & stores *claim review* from Guidewire Claimcenter
- V. Platform evaluates & initiates next step



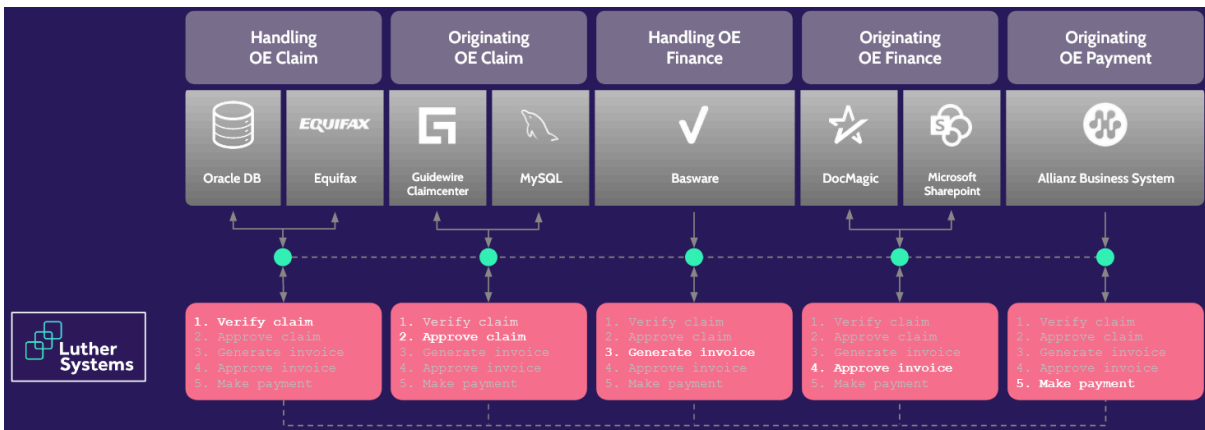
The steps operate in a similar manner until the final step is reached:

Step 8: Originating Payments team executes Make Payment, specifically Make Payment

- I. Platform sends *payment confirmation* (request) to Allianz Business System
- II. Platform receives *payment confirmation* (response) from Allianz Business System
- III. Platform validates *payment confirmation* based on predetermined rules in the Common Operations Script
- IV. Platform shares & stores *payment confirmation* from Allianz Business System
- V. Platform evaluates & initiates next step (process completion)



Final Step: The Platform completes the process:



10.2. Definitions

Term	Definition	Examples
Task	Simple events that are localized to one team involving one or two software systems	Copying data between systems, retrieving a claim from a database, making a payment
Workflow	A series of 10-20 tasks involving 1-2 software systems and 1-2 teams	Collecting related data from several systems, cross-checking claim data formats from different countries
Process	A series of 20+ tasks involving 3+ teams and multiple software systems	Settling a cross-department claim
Value Stream	A collection of processes delivering a business critical value	Claims
Participant	Operationally separate teams that have their own operations, governance and utilization of software systems and can make some autonomous decisions	Handling Claims Team, Originating Claims Team, Handling Finance Team, Originating Finance Team, Originating Payments Team
Team	As broadly defined by enterprises, otherwise known as departments, groups, units, etc.	All employees in the finance team at Allianz
Function	A unit of operations performed by a single team	Verifying a claim
Process Operations	End-to-end completion of process operations across multiple teams and software systems, to deliver a specific business objective	The end-to-end cross-department claims settlement process

10.3. Process Journey vs. User Journey

The Process Journey involves all the systems and teams including interactions with the users of the process, which usually interact with the process through UI systems and specifically designed Apps, with their own interfaces. However, process operations run through a much larger set of systems and teams, most of which are not visible to the user.

The User Journey is a small subset of the Process Journey. For an optimal User Journey, the whole process must operate reliably, not just the systems involved in the user journey! They must all operate correctly to operate the process end-to-end.

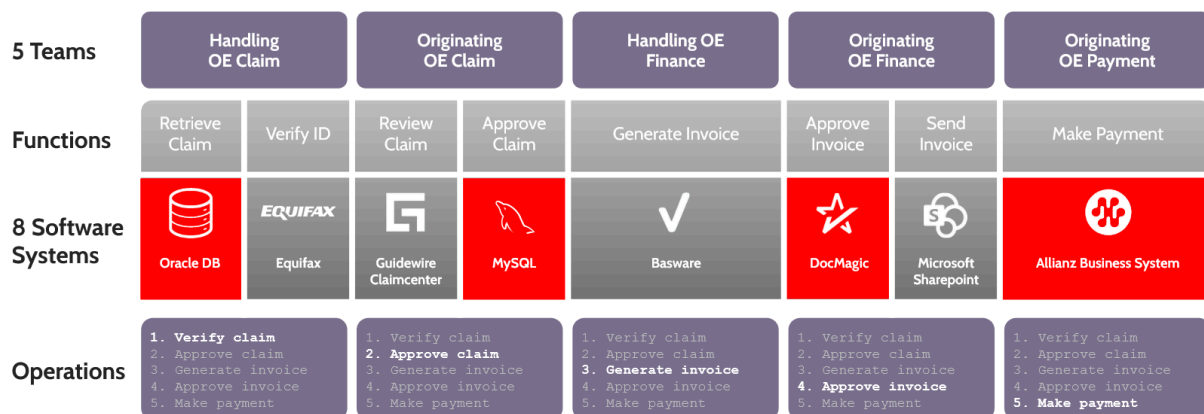


Fig 28. The process journey of the automated Cross-Department Claims Settlement Process. Systems highlighted in red directly interact with or require direct inputs from Allianz employees.

10.4. Plaintext Links

6. Implementation

For a walkthrough of the implementation process, view the Luther Systems Sandbox Setup:
<https://app.platform.luthersystemsapp.com/sandboxSetup>

For a full explanation of the implementation process, view the Full Luther Platform Setup:
<http://app.platform.luthersystemsapp.com>

For a more detailed description of the implementation steps please visit:
<https://www.luthersystems.com/platform/platform-overview>

Request access to an example of a more detailed timeline here:
https://docs.google.com/spreadsheets/d/1jHSeFRhaWVvkUiEtQ_crxGoyGFJ82eGUZ3rxhnYi4cro/edit?gid=1722375828#gid=1722375828

9. Luther's Company and Offerings

For more information about Luther's platform please visit our website:
<http://luthersystems.com>

Find out more about Luther's core platform features here:
<https://app.platform-test.luthersystemsapp.com/features>

For a demo of the build process please visit our website:
<https://app.platform-test.luthersystemsapp.com/build>

For more information about Luther's platform please consult this video:
<https://www.youtube.com/watch?v=78H5m1aZZoU>

For a more detailed introduction on the Luther platform and a full explanation of traditional process operations and Luther's solution please request access to the Deep Automation Primer here:
https://docs.google.com/document/u/1/d/103KIQUDuwMVOe5CziNFMYoYnq7g_7AoU_qIHLOza_Tw/edit

For a detailed introduction and documentation examples please see the Luther Platform site:
<https://www.luthersystems.com/platform/platform-overview>