Camunda at the heart of CERN's Electronic Document Handling (EDH) system

Dmitry Kekelidze

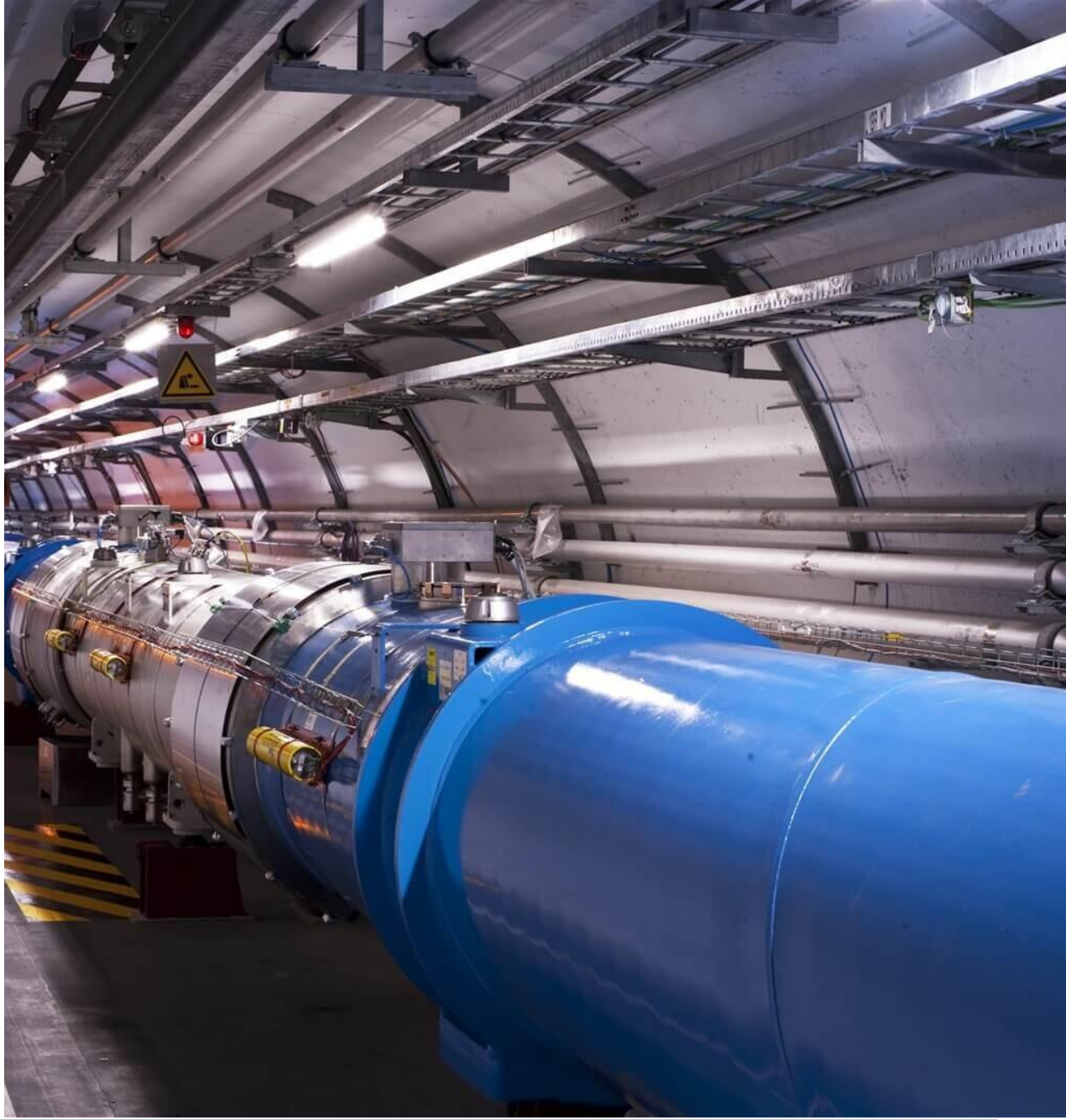
27.09.2023

What is CERN?

Conseil Européen pour la Recherche Nucléaire *(fr)*

European Council for Nuclear Research (en)

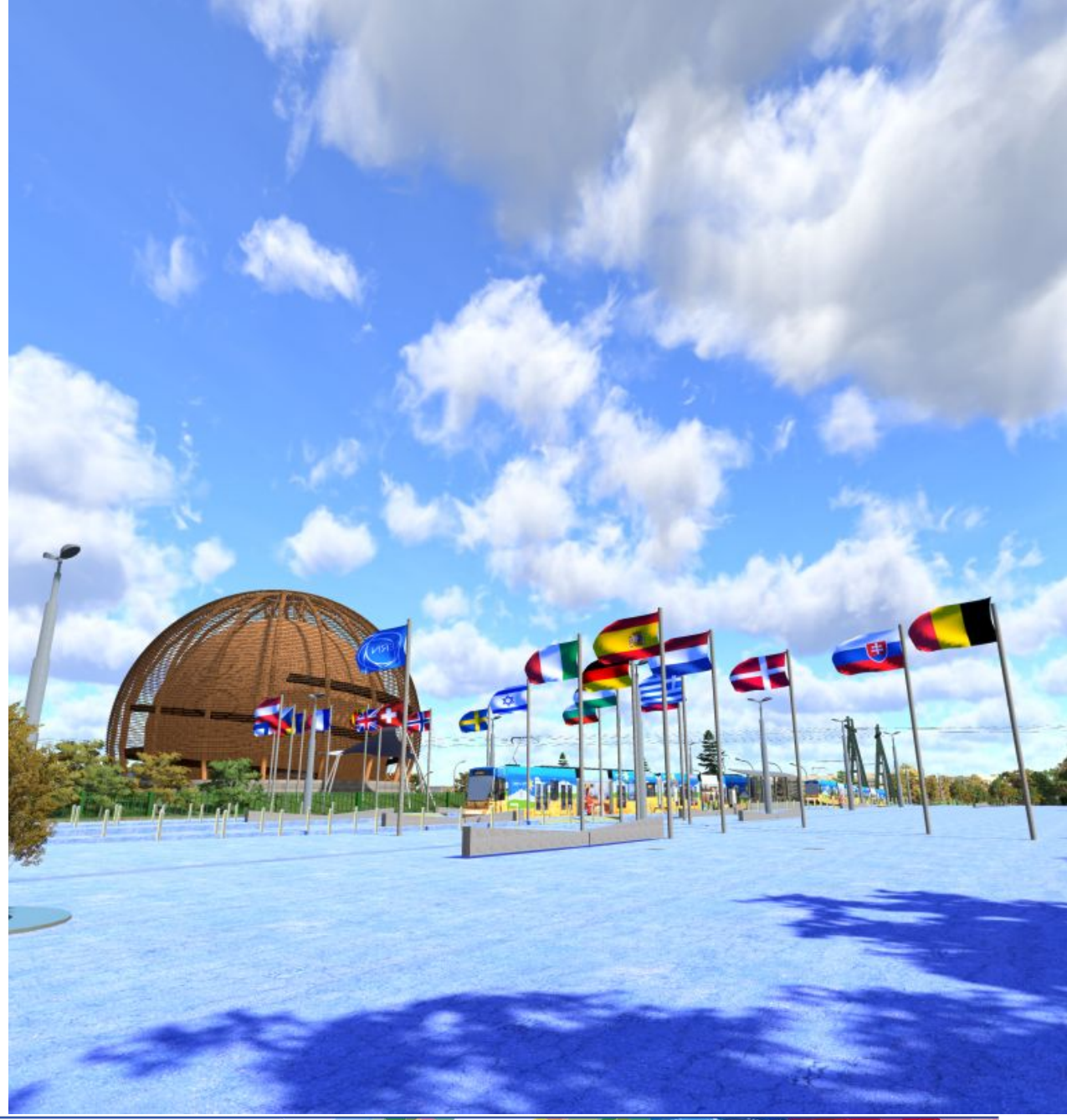
- Largest physics laboratory in the world with 9 active particle accelerators and a home to a wide range of experiments to carry out a diverse research program
- CERN Data Centre processes on average one petabyte (one million gigabytes) of data per day



What is also CERN?

People, infrastructure and money

- More than **17 500** people from around the world
- Buildings of all sizes covering a total footprint of **438 000** m2
- Yearly budget of over **1.2 Bn CHF**



What does that mean in terms of processes?

As any other organisation, CERN has many administrative processes and routines, such as

HR processes

Financial processes

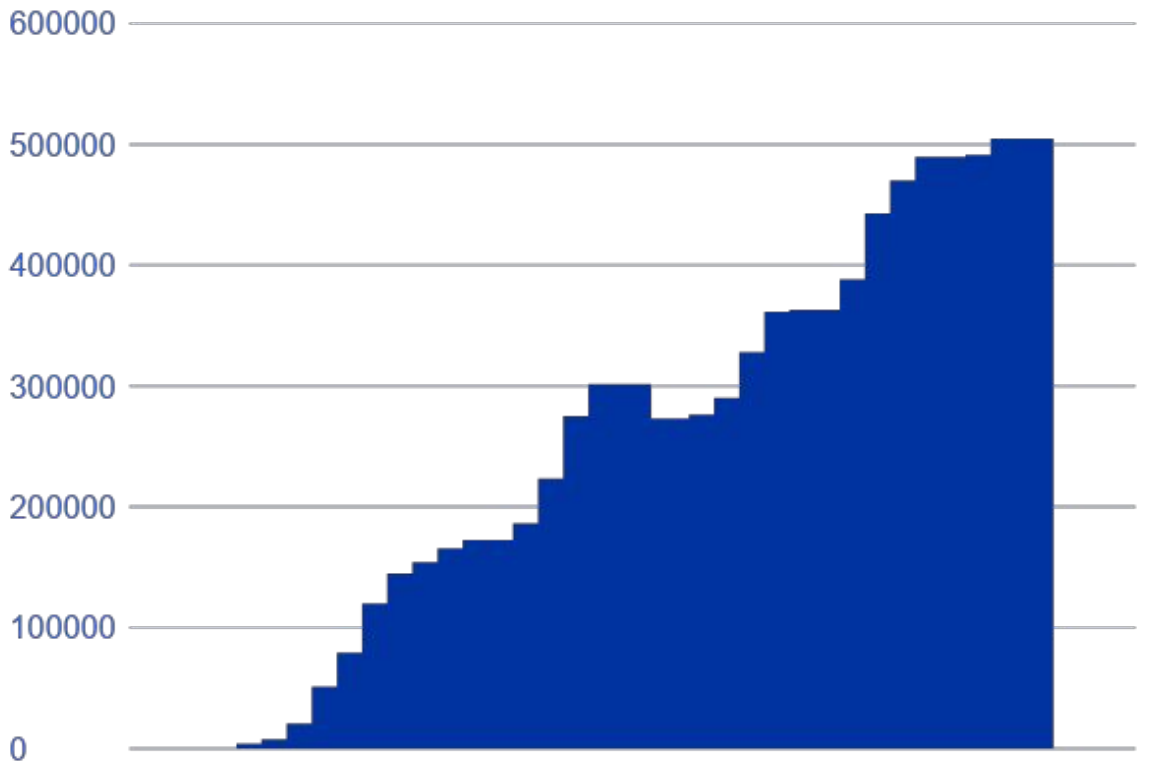
Facility management

and many more….

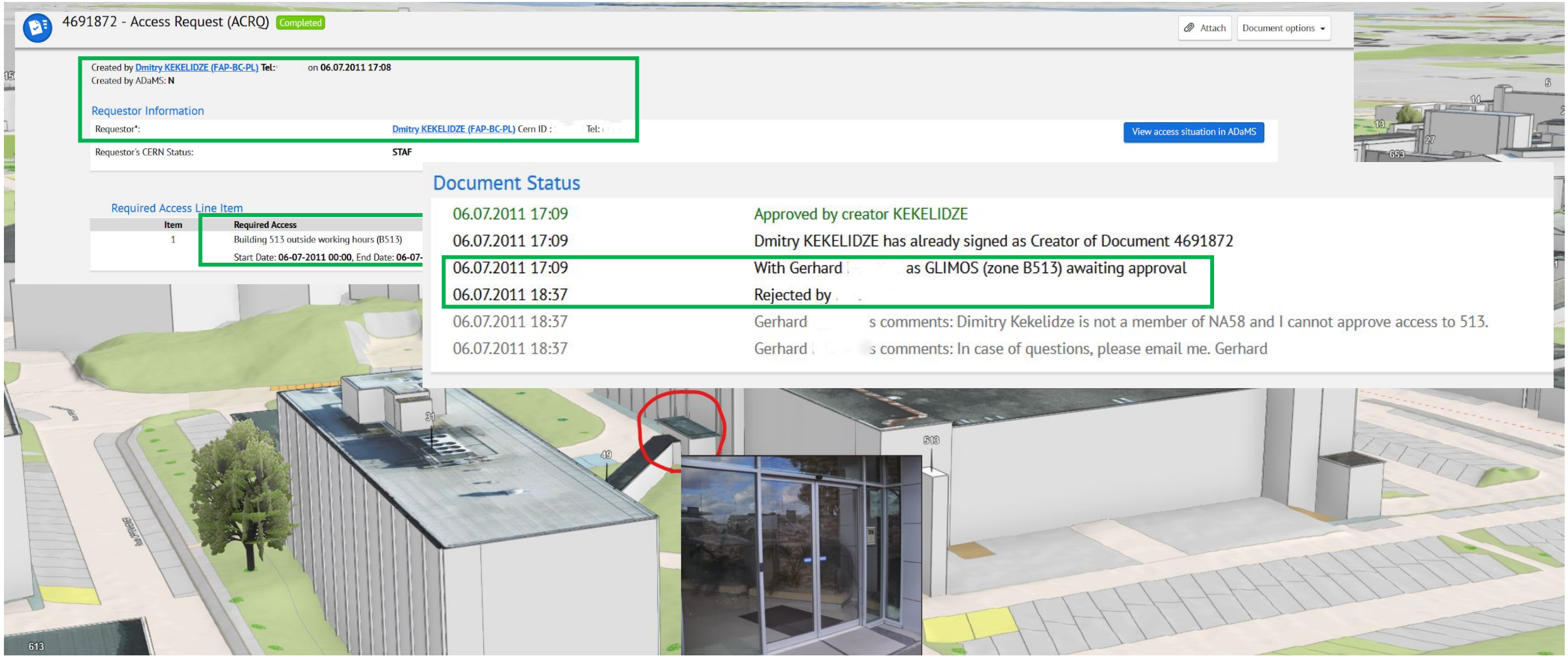
CERN administrative processes in numbers

Number of processes launched

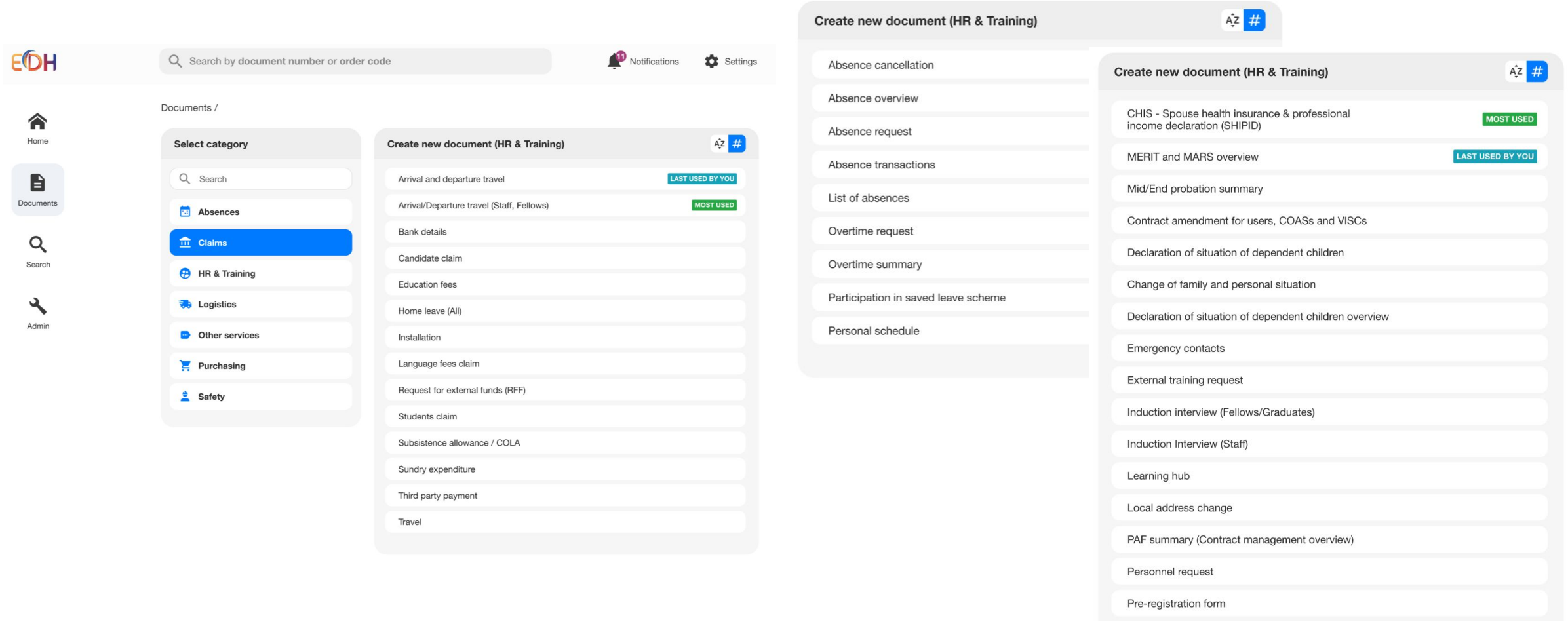
- **Since 1992 CERN was digitalising its administrative processes**
  - The Electronic Document Handling system (EDH)
- **By 2023 there is ~ 500’000 administrative processes executed digitally every year**



Requesting access to a building

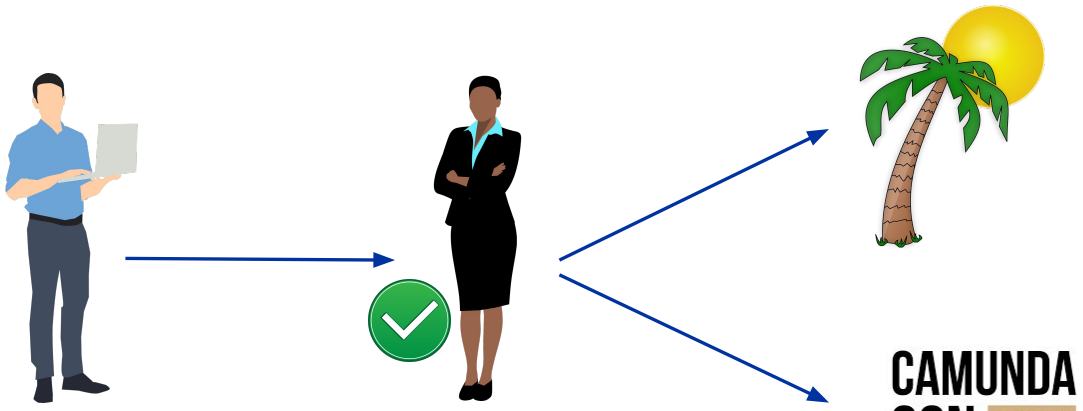


Variety of the processes

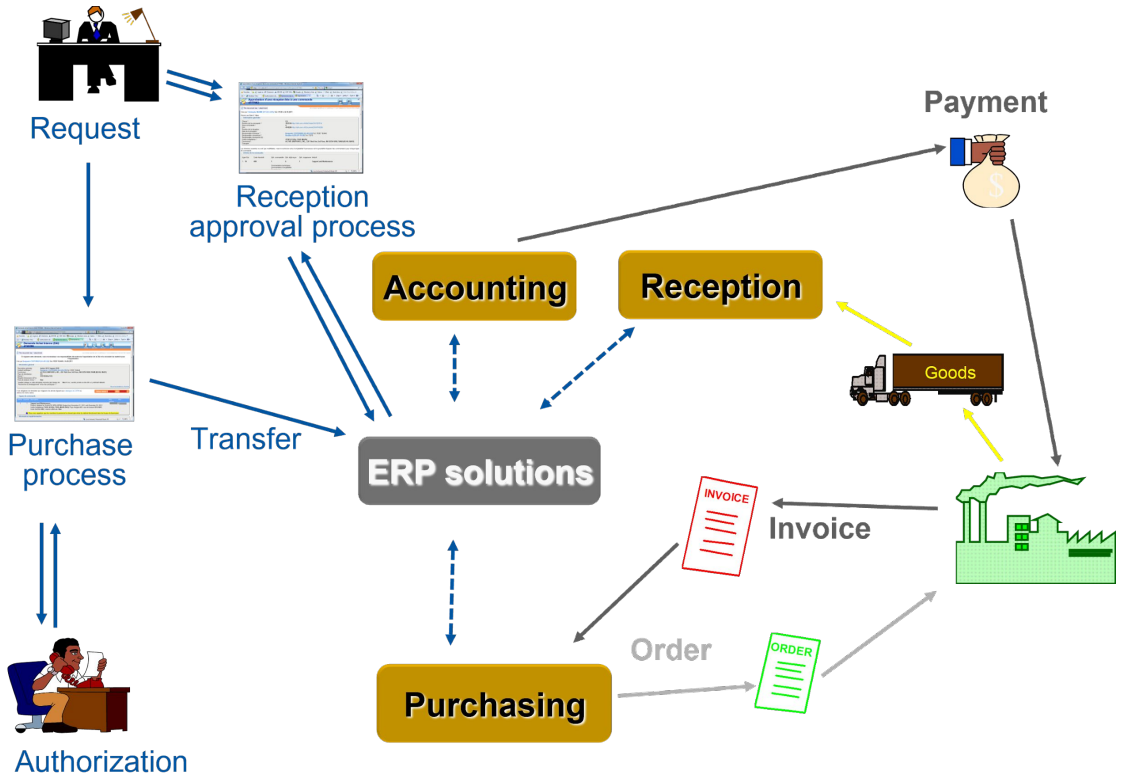


Complexity of the processes

Leave request



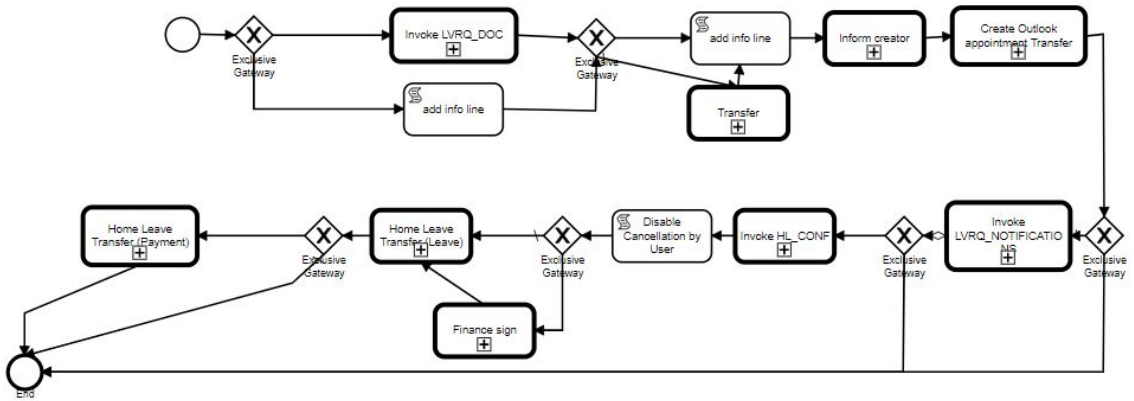
Purchasing Process Support

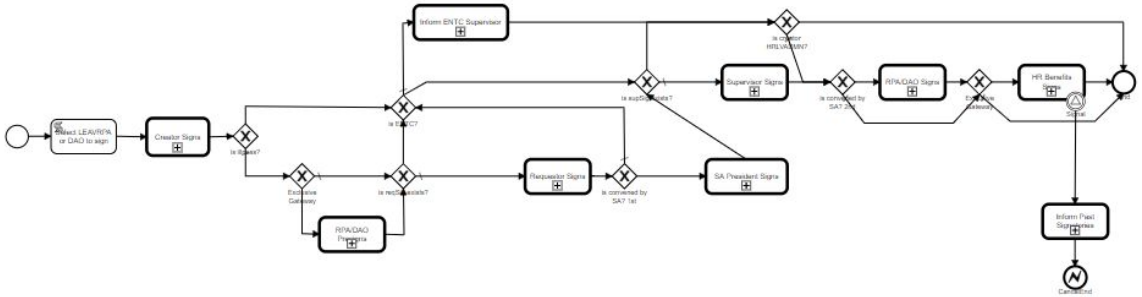


The “number one” process

Leave request

- **Total volume of roughly 2 million since it was introduced in 1992**
  - Holidays leave
  - Business trip
  - Telework
  - Medical leave
- **While being conceptually simple, it requires a complex diagram to define the process covering all possible scenarios**





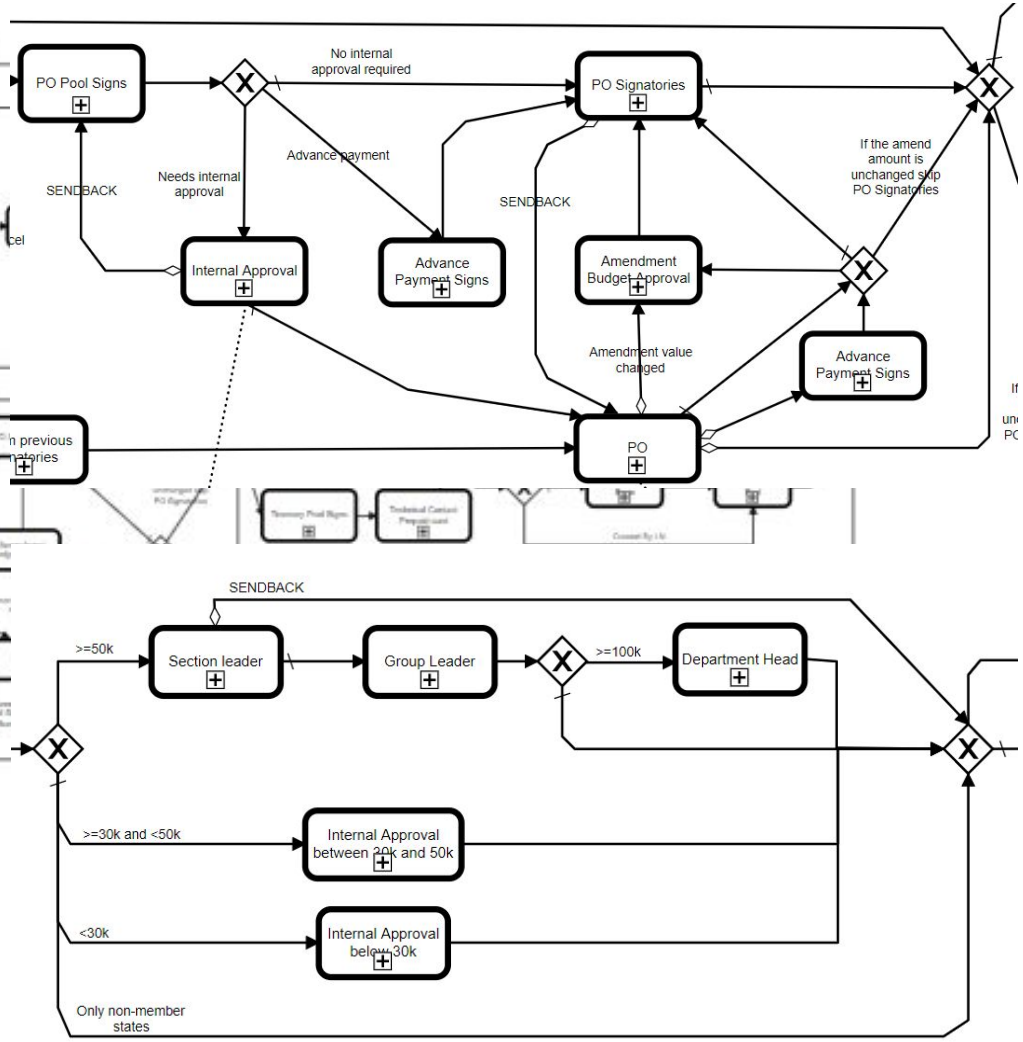
Purchasing process diagram

Complex administrative process

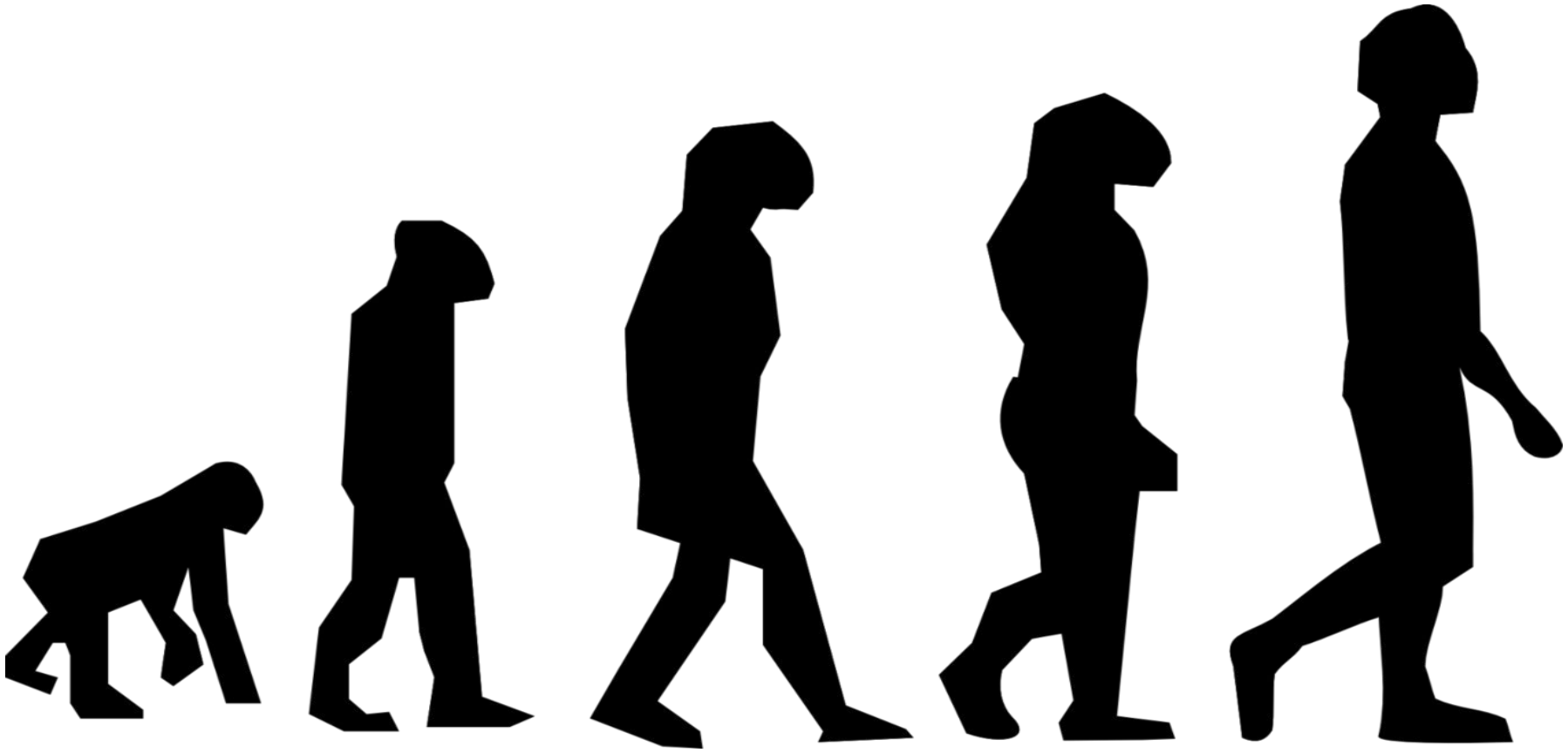
- Duration of some instances might reach 6 months
- Around 300 active process instances

Involves roughly 30 sub-processes

- Financial approvals based on the amounts
- Hierarchical approvals when multiple departments involved
- Integration with external systems



Evolution of workflow engines at CERN



1992	1998	2006	2013	2021
Self-made	*Oracle Workflow*	*ActiveVOS*	*Activiti*	CAMUNDA

Why Camunda?

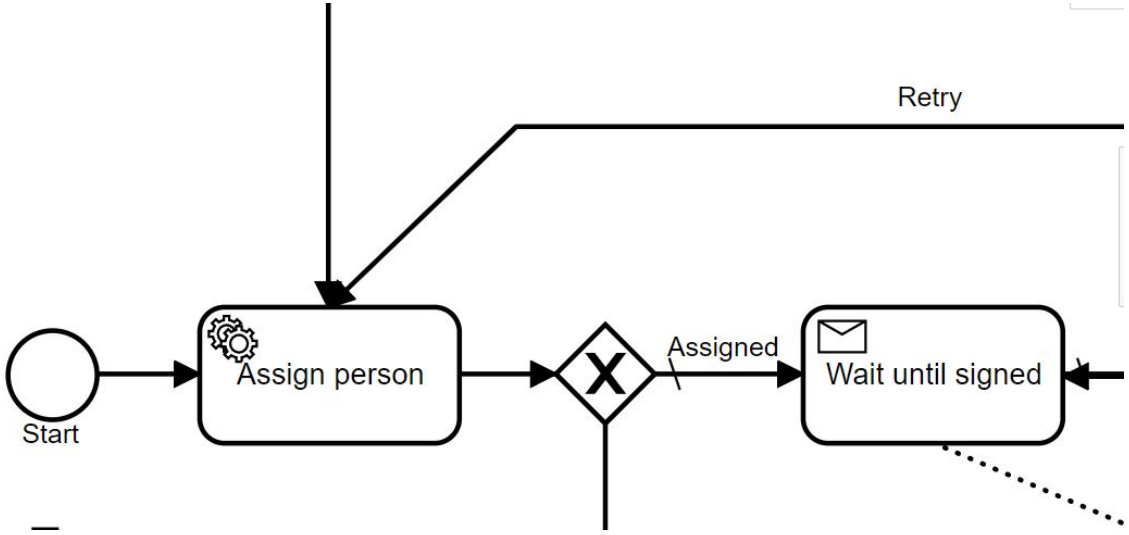
At the time the decision was taken, the following factors influenced it

- Camunda is a popular **actively supported and evolving** process orchestrator platform
  - BPMNs and 99% of the code can be migrated with little to no changes
  - Better **performance**
  - **Technology** upgrade
- Clear and exhaustive **documentation**
- Camunda **Cockpit**
- Camunda Modeler

Code that needed migration

The decision who should approve the process or be notified at each step is delegated to a “service task”

- Complex business logic that consists of 8 different “assignment rules” to choose the assignee
  - *~800 lines of code*
- Integration with CERN’s role-based access control system
- Integration with CERN external systems



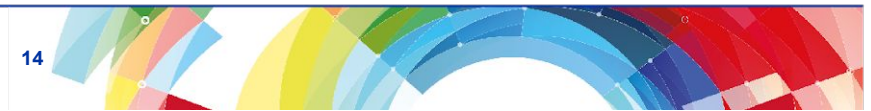
Migration speed

	Processes migrated	Pending
2021	7	
2022	38	
2023	20	
-		9

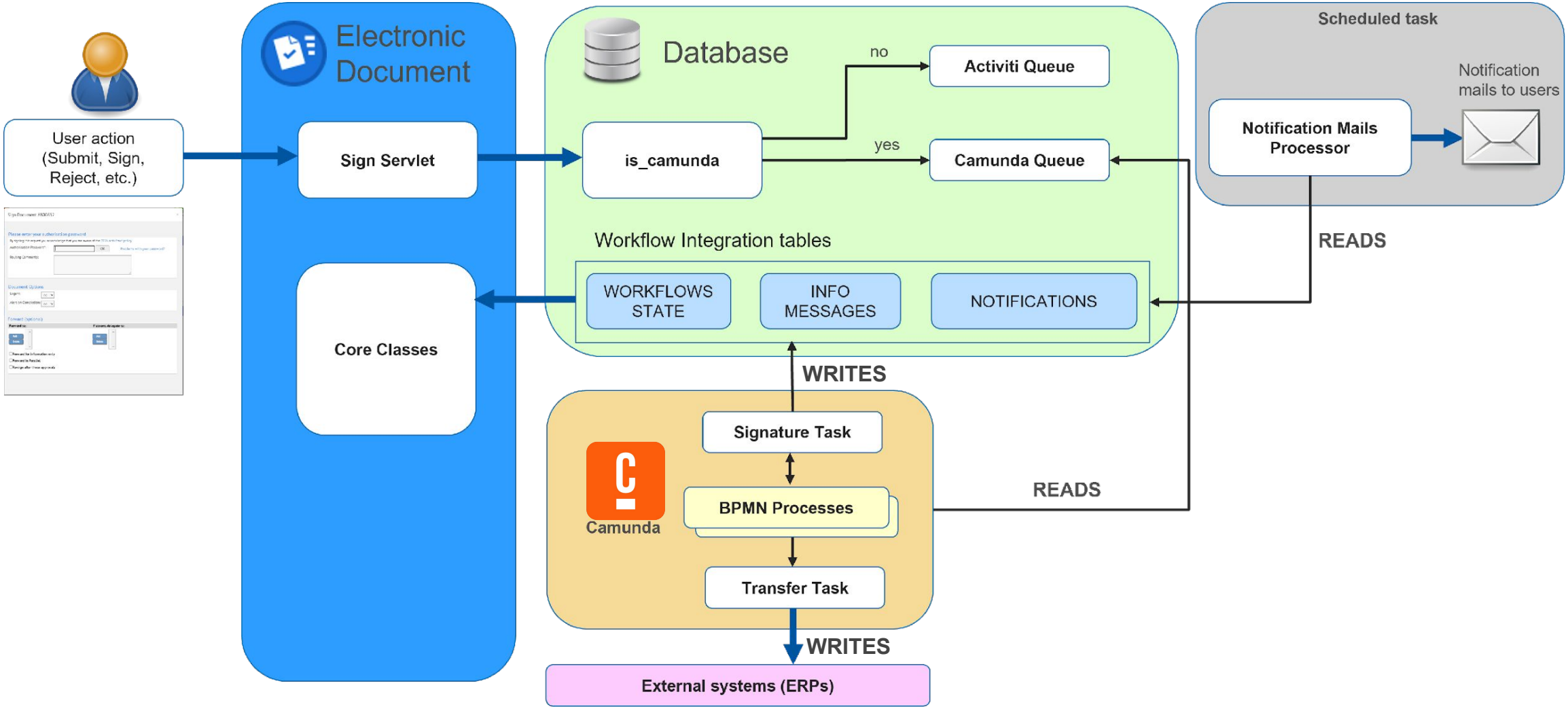
The migration started in 2021 by my team.

In one year, we migrated 7 processes, including the most complex purchasing process.

As of 2022, other teams migrate their respective processes after following a training given by my team.



Architecture to facilitate the migration



How to assure faultless migration?

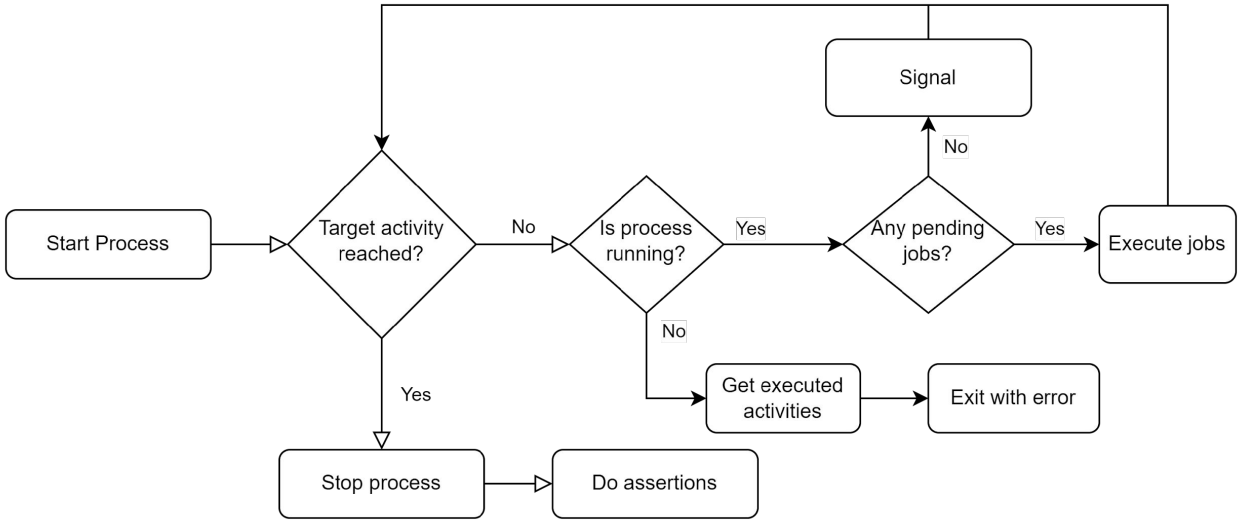
The complexity of the processes requires excessive testing

- Testing framework *(JUnit 5 based)*
  - Execute individual process from start to end
  - Advance the process until a given step
  - Mocks *(Mockito based)*
- Assertions
  - Steps executed
  - Process variables
  - Assignees selected
- H2 in-memory database for parallel tests executions

Test execution flow-chart

Test-case scenario

- Mock initial data.
- Start process execution for a given diagram.
- Signal the engine or execute jobs until the target activity is reached.
- Get historical process data and do assertions on it.

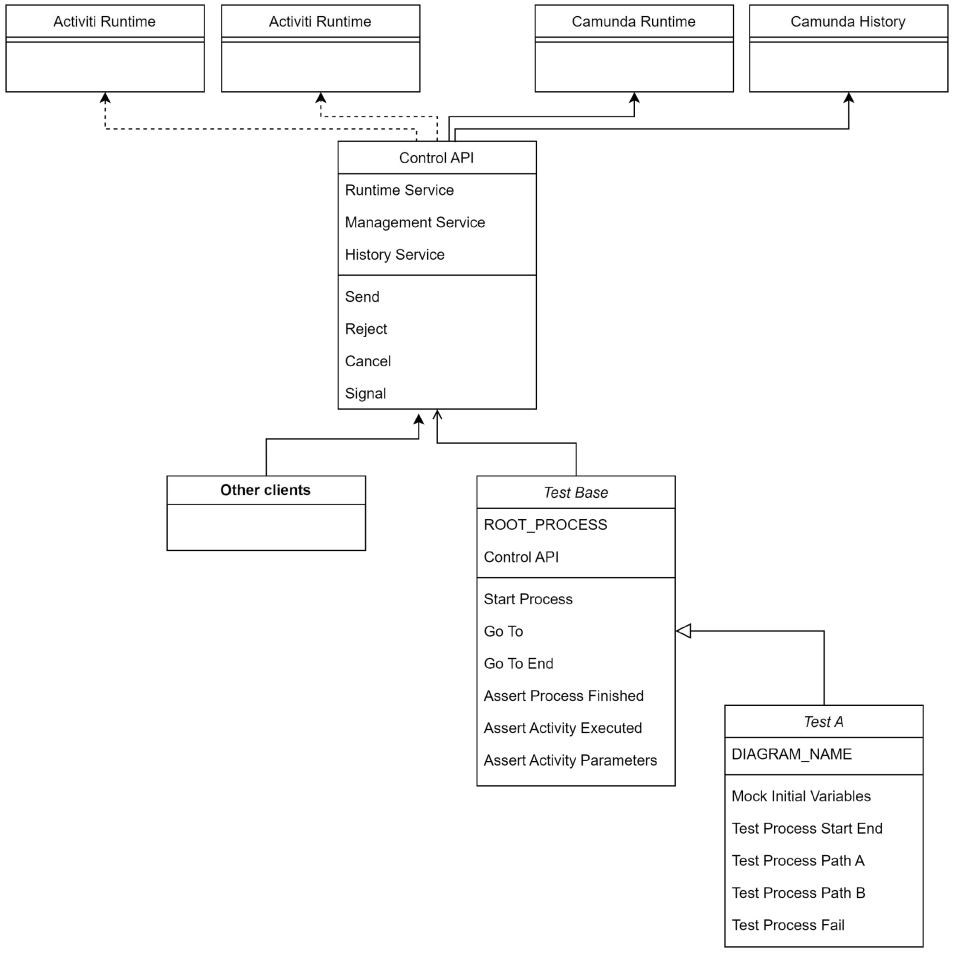


Bugs detected

- *Sub-processes that were not migrated.*
- *Bugs introduced during refactoring.*
- *Corner-cases of the decision flow.*

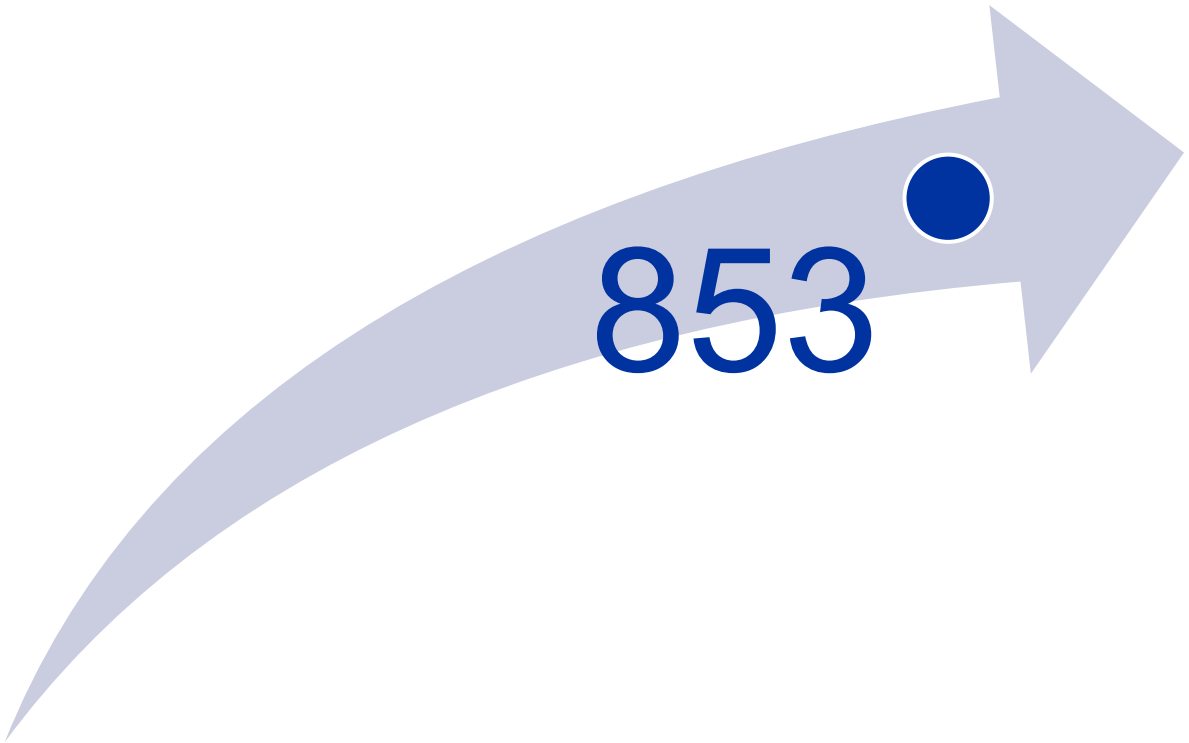
Test framework implementation

- API to interact with Camunda and Activiti runtimes through commands
- Test base class with control and assertion methods
- Individual tests extend the base class
- *Other clients interact with the same API*



Migration with confidence

Total number of tests



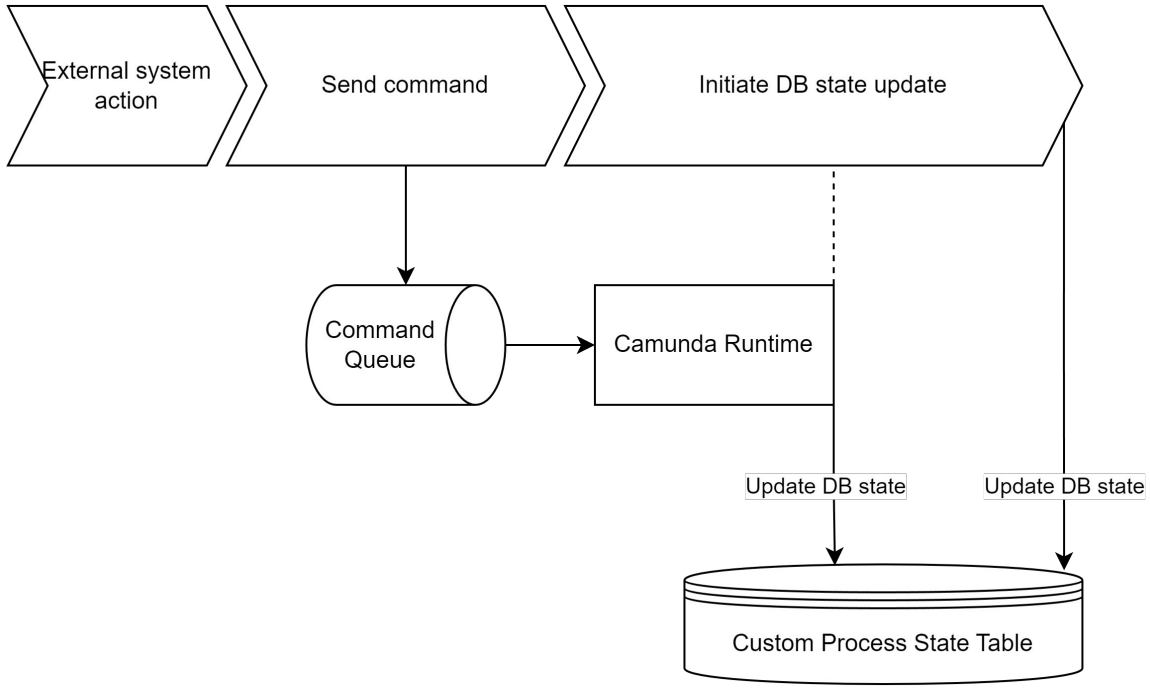
- 65 processes migrated to Camunda
  - **13** tests per process on average
- **No major incidents or regressions**

A fly in the ointment…

Despite the amount of automated and manual tests, we still encountered some problems during the migration

Camunda is “too fast”

- **Expose workflow engine to external systems**
  - Submits a command via an API
  - Updates the custom state in a table to prevent further commands submissions until workflow engine processes the event
- **Due to Camunda’s unprecedented speed, the queue event is processed faster than the initial DB state update**
  - Results in an inconsistent state

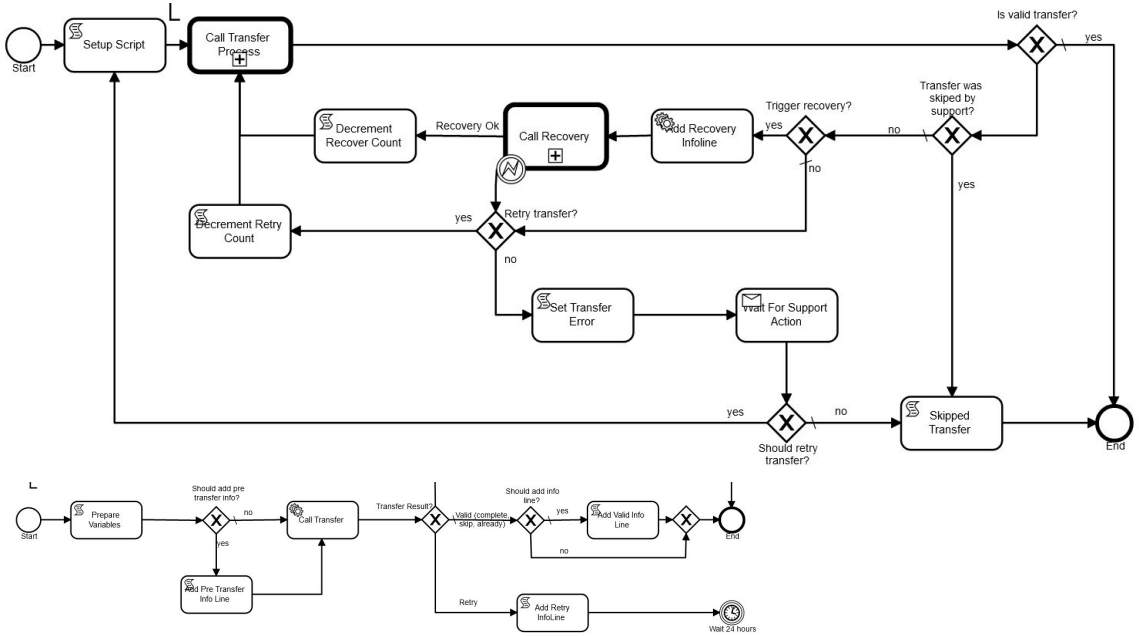


Rely on the Camunda runtime/historical data instead of the custom state table.

Don’t just copy and paste

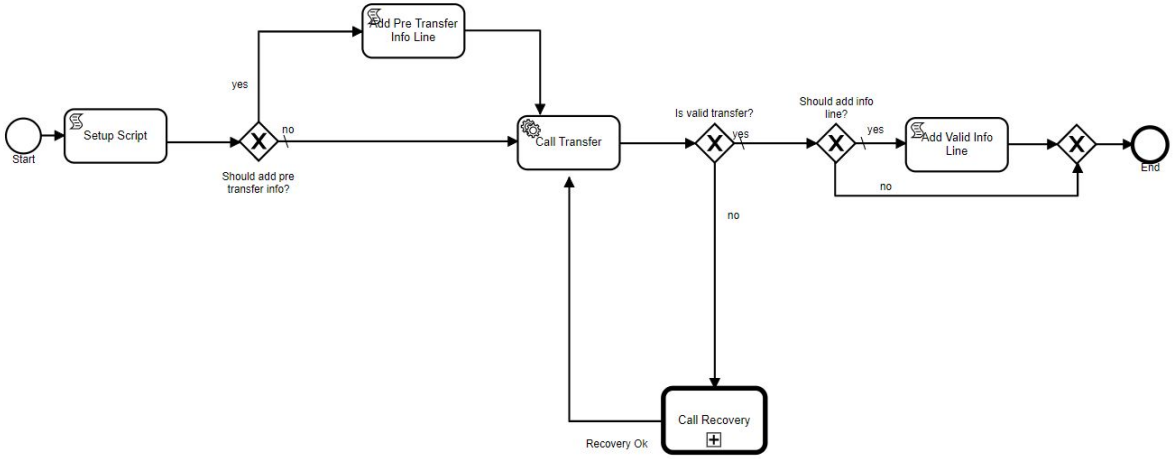
Before

- Overcomplicated transfer process diagram
- Recovery handled by a custom Java code



After

- Simple straight-forward diagram
- Exceptions and retries are handled by Camunda



What did we gain with Camunda?

- **A very fast and stable process orchestration engine**
  - Zero overhead during event processing
  - The JVM process instance requires no interventions between deployments
- **Modern technology stack**
  - Suitable for GitOps/K8s
  - Easy to run locally
- **Automated recovery/incidents creation for service tasks**
- **Powerful Cockpit** *(appreciated by the quality assurance and support teams)*
- **Well documented** *(newcomers get on-board quickly)*

Next steps

- **Migrate the rest of our processes to Camunda**
  - 9 processes left
- **Provide additional Camunda test environments for the product groups**
  - Take advantage of the K8s infrastructure
- **Enhance Camunda Cockpit with custom plugins**
- **Upgrade to Camunda 8**
  - Requires additional work to expose external data via REST
  - *Once all existing processes are migrated*


DANKE!
THANK YOU!
MERCI!
GRAZIE!
GRACIAS!
DANK JE WEL!

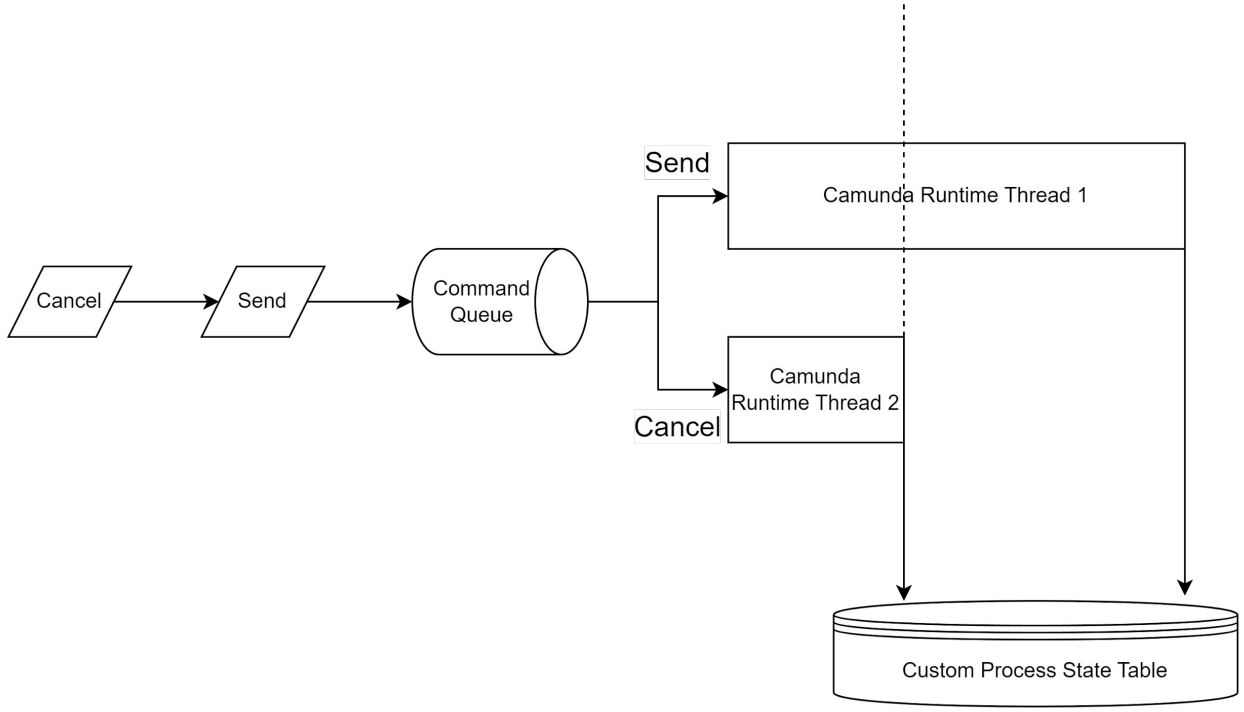
Test code examples

```
@Test
public void testDAIAmendmentSignOrder() {
  DAI dai = mockDai();

  Mockito.when(dai.isDAIAmendment()).thenReturn(true);
  Mockito.when(dai.getAmendedDocumentId()).thenReturn(testDocId + 6);

  doReturn(99000d).when(dai).getTotalCHF();
  doReturn(false).when(exclusionService).isPersonExcluded(any(), any(),
eq(false));

doReturn(false).when(daiWorkflowService).hasAmendmentTotalRemainedUnchanged(any
());

  doReturn(Collections.singletonList("$" +
mockUtils.GYRO_GEARLOOSE().getPersonId()))
      .when(interactionApi).getCompleteAndInformSignatories(any());

  mockRight("SL", mockUtils.NORTON_NIMNUL());
  mockRight("PC", mockUtils.MICKEY_MOUSE());

  String pid = startDaiProcess();

  goToEnd(testDocId, pid);

  assertTargetActivities(pid, "Inform previous signatories", "PO", "Amendment
Budget Approval",
      "PO Signatories", "DAI Transfer", "End");
}
```

```
@Test
public void testBankInformCreator() {
  var creator = mockUtils.ANDREW_SHORT();
  var doc = mockDocument(creator);
  var processId = startProcess(BANK_PROCESS, testDocId);

  var actual = goToSignature(testDocId, processId, "Inform Creator");
  var expected = new SignatureParameters();
  expected.setDocumentId(testDocId);
  expected.setSignature("Creator");
  expected.setSignatureType(SignatureParameters.INFORMATION);
  expected.setEmail("infoCompleteMessage.xml");
  assertSignatureEquals(expected, actual);

  checkLastInfoLineContains(testDocId,
      "Informing " + creator.getFullName(),
      "Informe " + creator.getFullName());

  assertEquals("infoCompleteMessage.xml",

getSentNotificationParameters().get(getSentNotificationParameters().size() - 1)
          .getMailTemplate());
}
```


Multi-threading issue

- **Commands processing is asynchronous**
  - Commands are inserted in a queue sequentially
  - Commands are consumed from the queue asynchronously
- **Inefficient query to Camunda historical process data lead to some commands being processed slower than the others**
  - Results in an inconsistent state



Optimize the historical data query - use indexed business key as the query condition.

“Text and four pictures” slide

Heading

- Maecenas at maximus augue, sed ultrices urna. Nunc non dignissim arcu. Nunc ut efficitur diam.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Heading

- Maecenas at maximus augue, sed ultrices urna. Nunc non dignissim arcu. Nunc ut efficitur diam.
- Lorem ipsum dolor sit amet, consectetur adipiscing elit.

“Subtitle and three pictures” slide

Subtitle

Subtitle

Subtitle