



TEMPORAL: SWEET THEORY

Speaker: Jevgenij Liachovič @ Ovoko
<https://www.linkedin.com/in/jevgenijliachovic/>



TEMPORAL: SWEET THEORY



**USE
CLASSIC
SOLUTION**



**USE
SOMETHING NEW
AND
LESS KNOWN**

Speaker: Jevgenij Liachovič @ Ovoko
<https://www.linkedin.com/in/jevgenijliachovic/>



Today's Agenda



Today's Agenda





Today's Agenda

A

Use case
presentation



B

(Not so) Simple
use case
implementation
without Temporal





Today's Agenda

A

Use case
presentation



B

(Not so) Simple
use case
implementation
without Temporal



C

Simple use case
implementation
with Temporal





Today's Agenda

A

Use case presentation



B

(Not so) Simple
use case
implementation
without Temporal



C

Simple use case
implementation
with Temporal



D

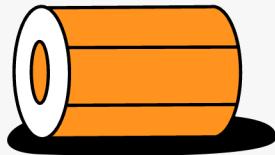
Working POC
review





How we learn?

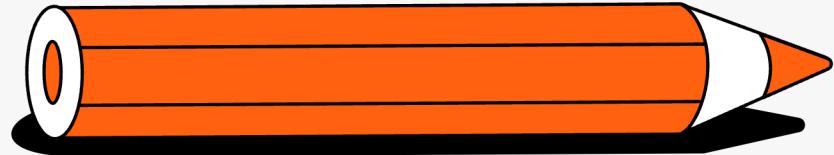
Theory



Practise



Mistakes !

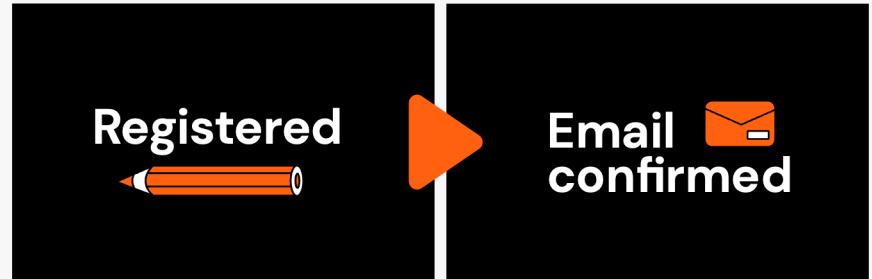




A Solve issue: small service simple issue

Example: Registration

Solution: Entity + State constants

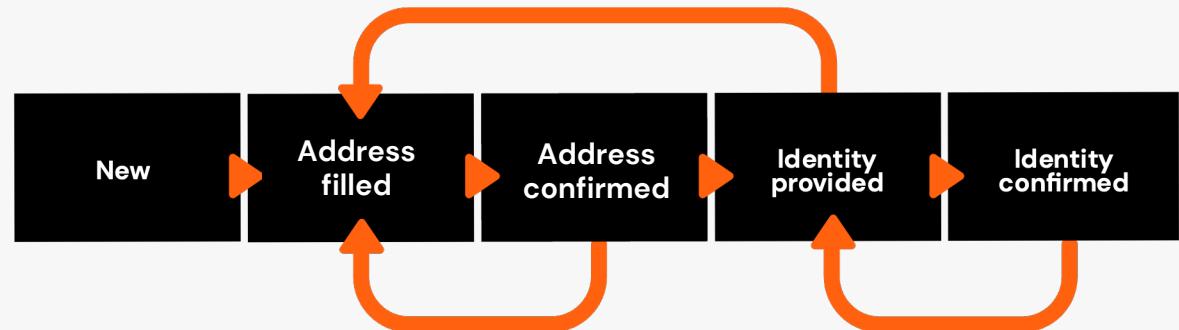




A Solve issue: bigger service, complex issue

Example: Submission of application

Solution: Entity + states constants + State machine/workflow





A Solve issue: distributed system, complex system

Example: Order processing

Solution:

- Event based architecture (+ Kafka)
- DAG (Directed Acyclic Graph) solution
 - Netflix Conductor
 - Apache Airflow
- Temporal



A What is a **workflow**?

01

Wikipedia



A workflow consists of an orchestrated and repeatable pattern of activity, enabled by the systematic organization of resources into processes that transform ... or process information

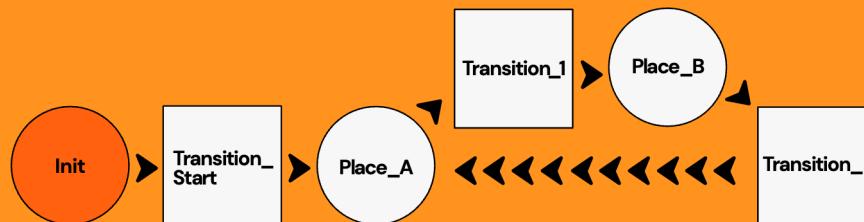


A What is a workflow?

02 Symfony



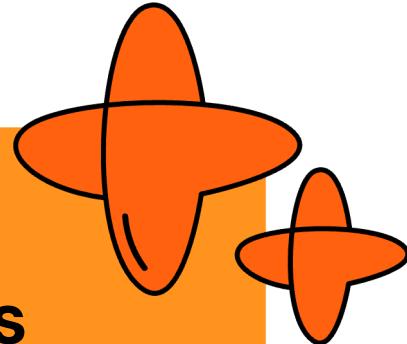
... way to define a process or a life cycle that your object goes through...





A What is a **workflow**?

O3 **Temporal Founders**

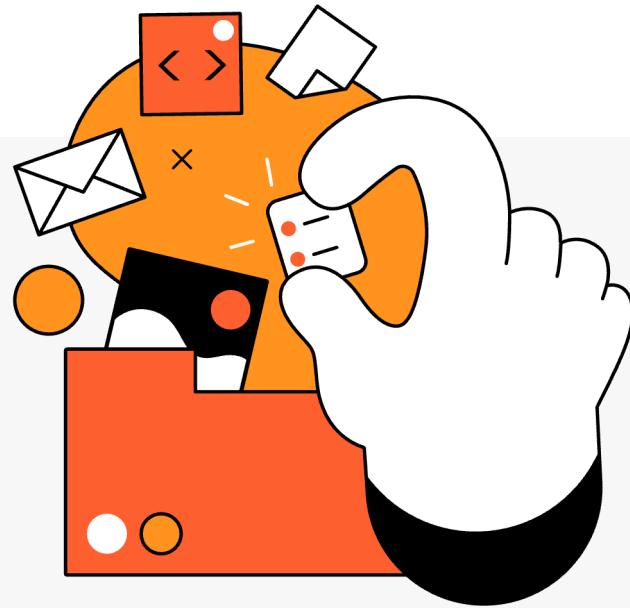


- Sequence of tasks with coordination logic
- Durable and observable application
- Reacts to external events
- Timers and timeouts



A Workflow requirements: observable

- «What is the current state?»
- «Read execution result»
- «Audit logs»
- «Workflow lookup (search)»





A Workflow requirements: durability

- «Manage errors correctly»
- «Timeouts»
- «Retries»
- «Audit log»
- «Stack traces»





A Use case for workflows

- O1 Term of use
- O2 Submission of application
- O3 Subscription
- O4 Order processing
- O5 Loan payout
- O6 etc.



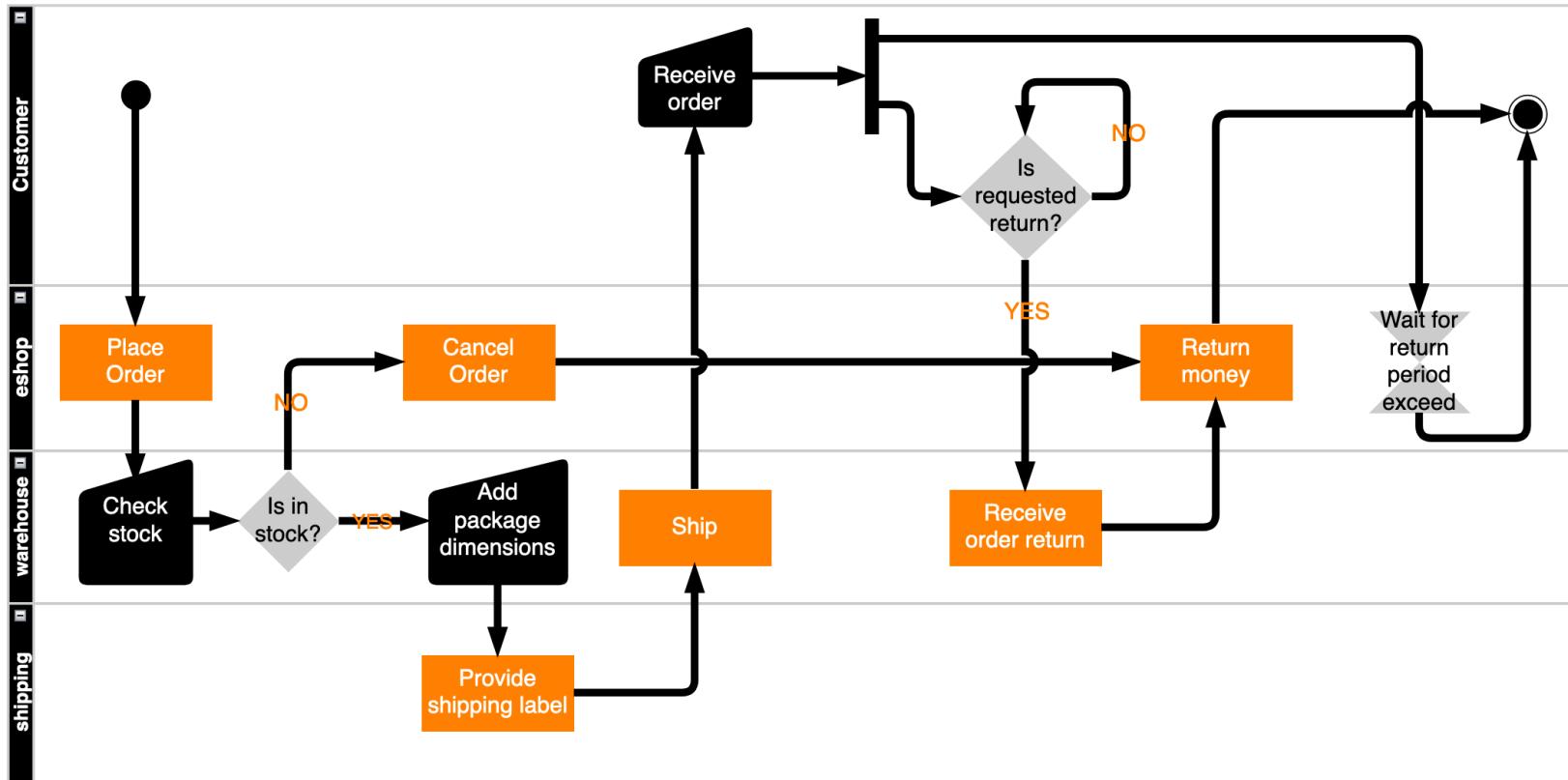
A Use case for workflows

- O1 Term of use
- O2 Submission of application
- O3 Subscription
- O4 Order processing
- O5 Loan payout
- O6 etc.



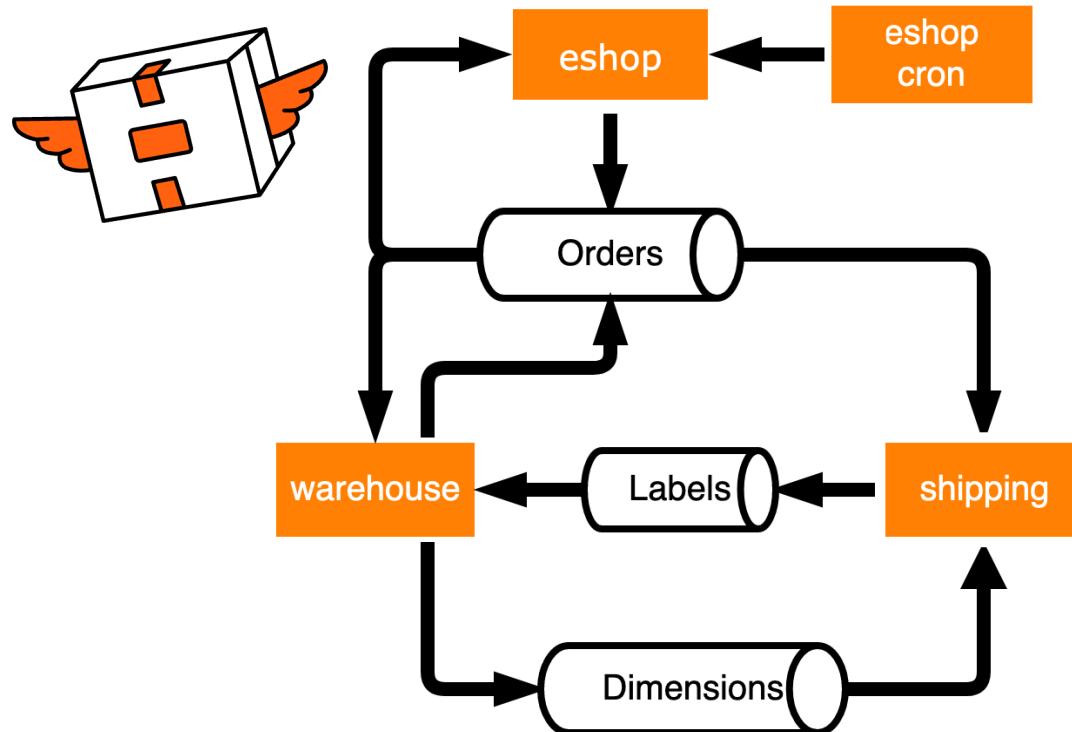


A Use case to discuss: order processing



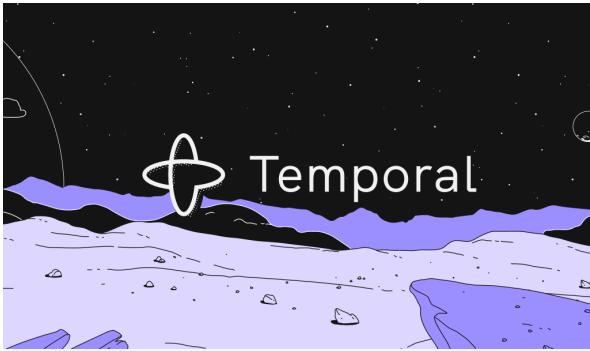


B Possible solution for order processing

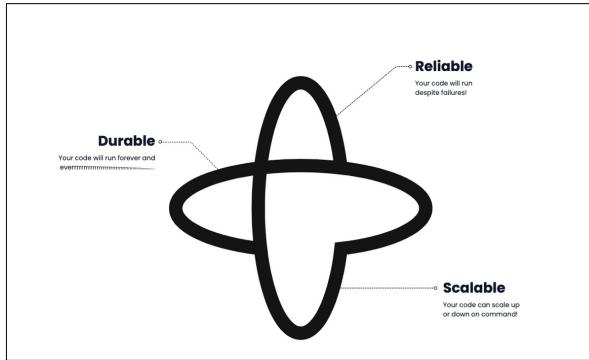




C Temporal:



Platform to build
applications for managing
complex workflows and
orchestrating distributed
systems



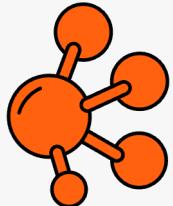
Fault-tolerant and durable workflow engine that ensures the execution of **critical business logic** and **coordination** of various components within an application



C Temporal: durability



C Temporal: durability

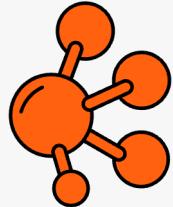


Transactional outbox

transfer queue +
sharding
(to ensure ACID)



C Temporal: durability



Transactional outbox

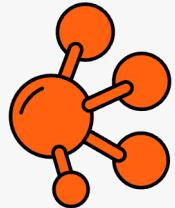
transfer queue +
sharding
(to ensure ACID)



Atomic event history (audit-log)



C Temporal: durability



Transactional outbox

transfer queue +
sharding
(to ensure ACID)



Atomic event history

(audit-log)



Recovery from any point of time



C

Temporal – *observability*



C Temporal – *observability*

- Metrics



C Temporal – *observability*

- Metrics
- Tracing



C

Temporal – *observability*

- Metrics
- Tracing
- Logging



C

Temporal – *observability*

- Metrics
- Tracing
- Logging
- Visibility



C

Temporal – observability

- Metrics
- Tracing
- Logging
- Visibility
- UI

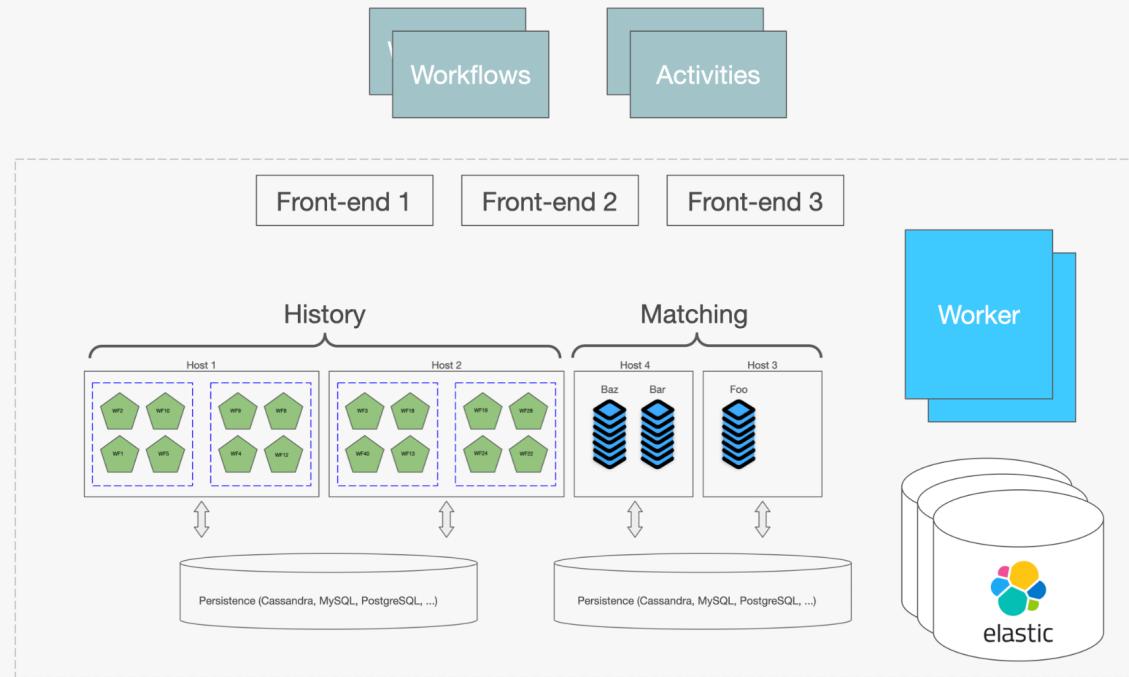


The screenshot shows the Temporal web interface version 1.15.0. The top navigation bar includes the Temporal logo, a namespace dropdown set to "default", and tabs for WORKFLOWS, SETTINGS, and ARCHIVAL. A "Report Bug/Give Feedback" button is also present. Below the navigation is a search bar and filter controls for Workflow ID, Workflow Name, Status, Start Time (set to "Last 30 days"), and End Time. An "ADVANCED" button is available for more filtering options. The main table lists workflow runs with columns: WORKFLOW ID, RUN ID, NAME, STATUS, START TIME, and END TIME. The data shows several completed runs for the OrderWorkflowInterface across different run IDs and timestamps.

WORKFLOW ID	RUN ID	NAME	STATUS	START TIME	END TIME
901957fd-f215-47f9-9ffb-8172a98ceaab	f95653cd-9862-498e-8899-535796b53ca5	OrderWorkflowInterface	Completed	May 31, 2023 1:01 AM	May 31, 2023 1:03 AM
6fdd5ad4-132f-45a1-9b3a-4423d7aa61ea	bfa66d6c-3614-4714-971d-1165ddbe3f0d	OrderWorkflowInterface	Completed	May 31, 2023 12:57 AM	May 31, 2023 12:59 AM
60aa74cf-512b-453d-b86a-e1617a165966	13b4081d-7728-40ee-8bcd1-cd475d52ff6f	OrderWorkflowInterface	Completed	May 31, 2023 12:48 AM	May 31, 2023 12:50 AM
613307e8-1583-435c-993e-7fde362be283	97291b57-9e28-47c4-a517-599724577962	OrderWorkflowInterface	Terminated	May 31, 2023 12:40 AM	May 31, 2023 12:47 AM
b05cf2ac-4ad0-460c-90d8-59056c3b4ee3	a51a1b31-a2df-4432-8568-8b3497fb5d06	OrderWorkflowInterface	Completed	May 31, 2023 12:32 AM	May 31, 2023 12:34 AM
d271ef77-9de4-4cbb-af9c-4bcba9e3141f	06251ad9-26ee-4b24-ae04-ec6e415e4af8	OrderWorkflowInterface	Completed	May 31, 2023 12:25 AM	May 31, 2023 12:30 AM
-994e-4a0f-b7ac-83fa8b551c04	02abcb3e-2ebc-404b-b140-ea05b9cab61d	OrderWorkflowInterface	Completed	May 30, 2023 11:10 PM	May 30, 2023 11:12 PM
xe-ed7b-4073-aa82-2588ec230753	fc98cdfe-834d-48e4-8206-e0d8470d0636	OrderWorkflowInterface	Failed	May 30, 2023 10:45 PM	May 30, 2023 10:48 PM



C Temporal overview





C Temporal overview



C Temporal overview

O1 Workflow

- Stateful
- Asynchronous codebase,
coroutines
- Fault tolerant process
- Deterministic
- No I/O operations
- Timers for any time
- Child workflow



C Temporal overview

01

Workflow

- Stateful
- Asynchronous codebase, coroutines
- Fault tolerant process
- Deterministic
- No I/O operations
- Timers for any time
- Child workflow

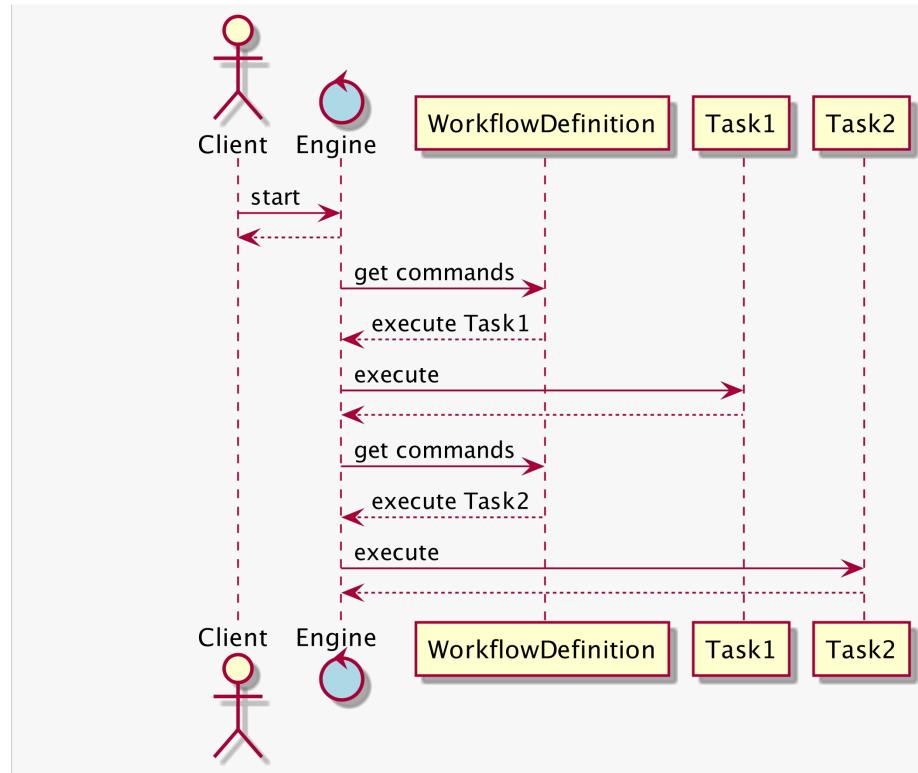
02

Activity



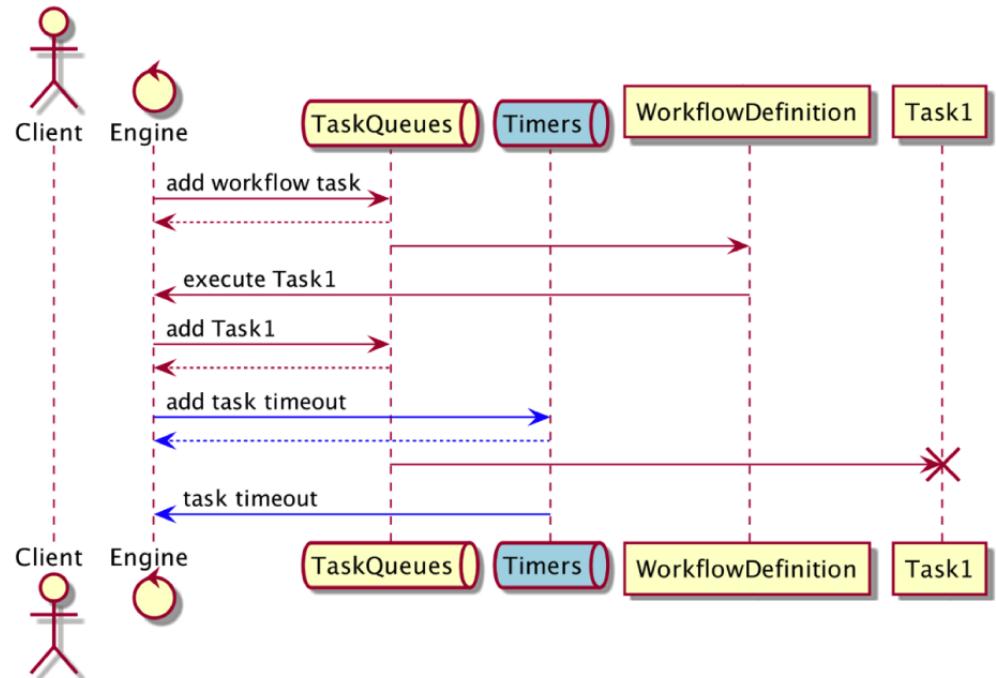
- Stateless
- Regular codebase
- Any tools
- Any languages (PHP, Go, Java)
- Fails are OK

C Temporal workflow

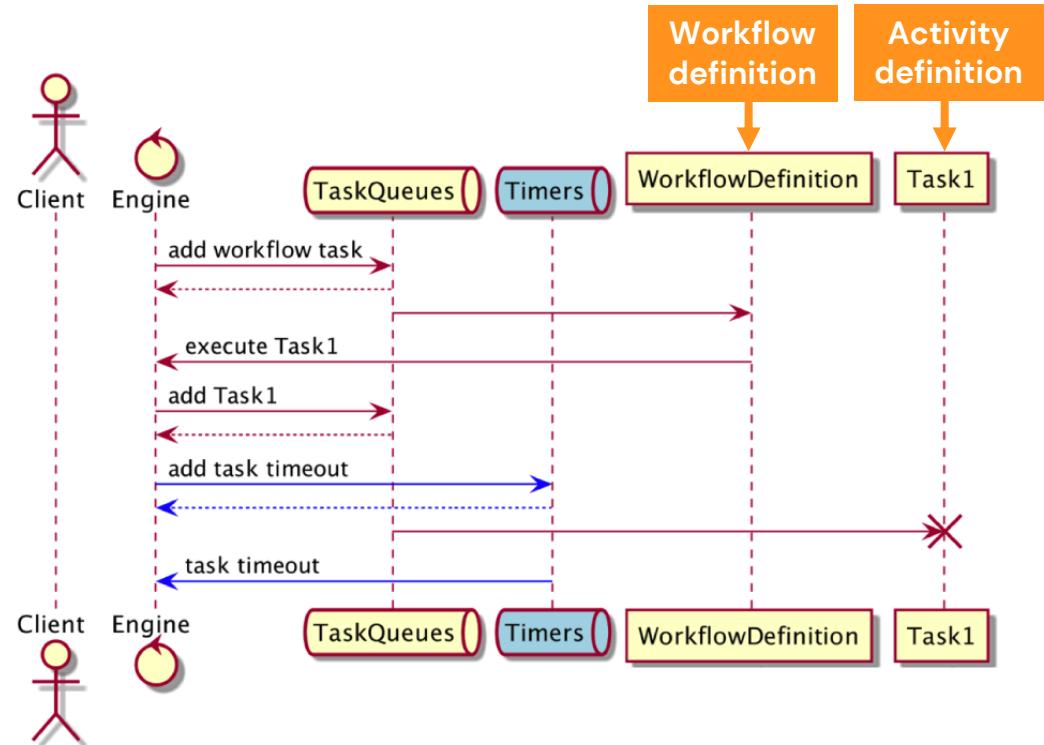




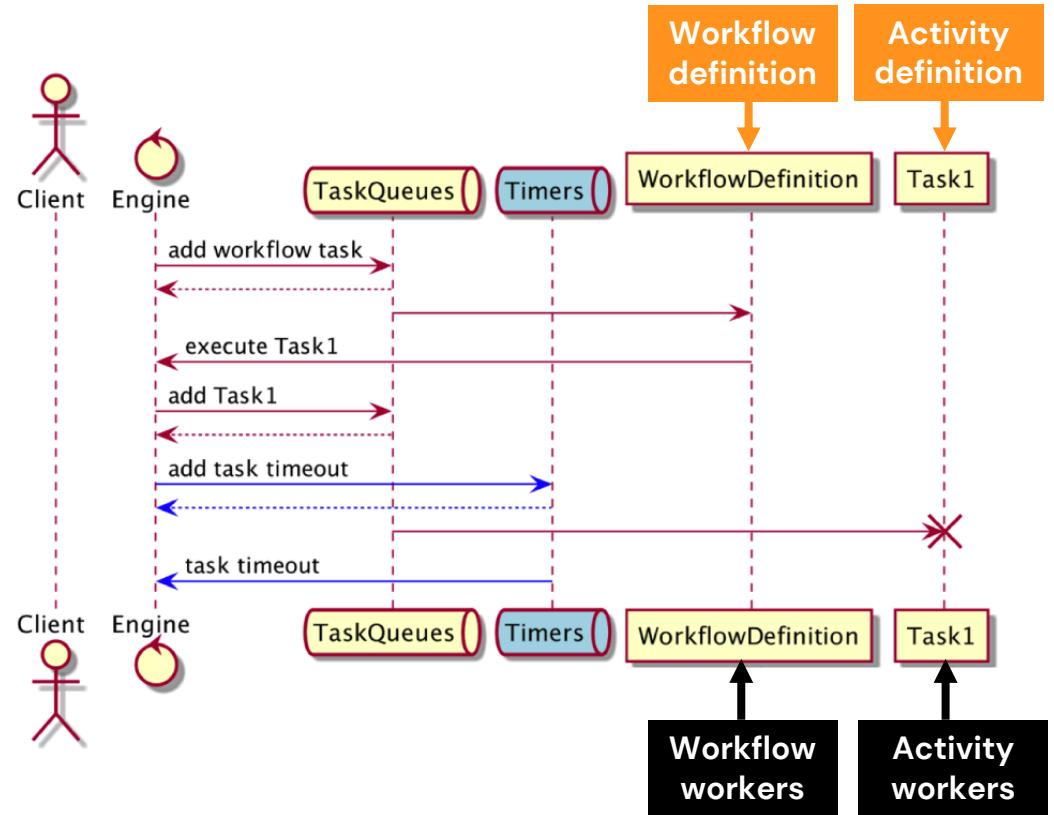
C Temporal workflow



C Temporal workflow

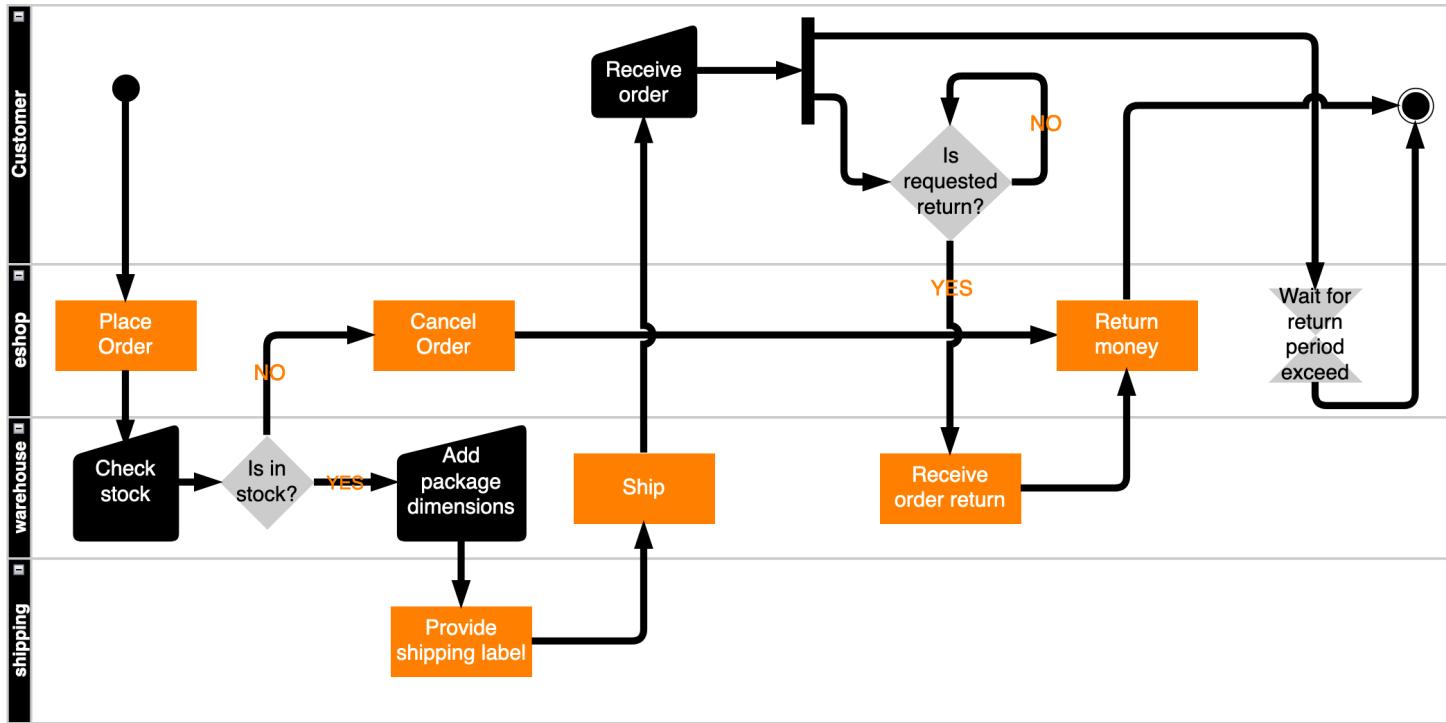


C Temporal workflow



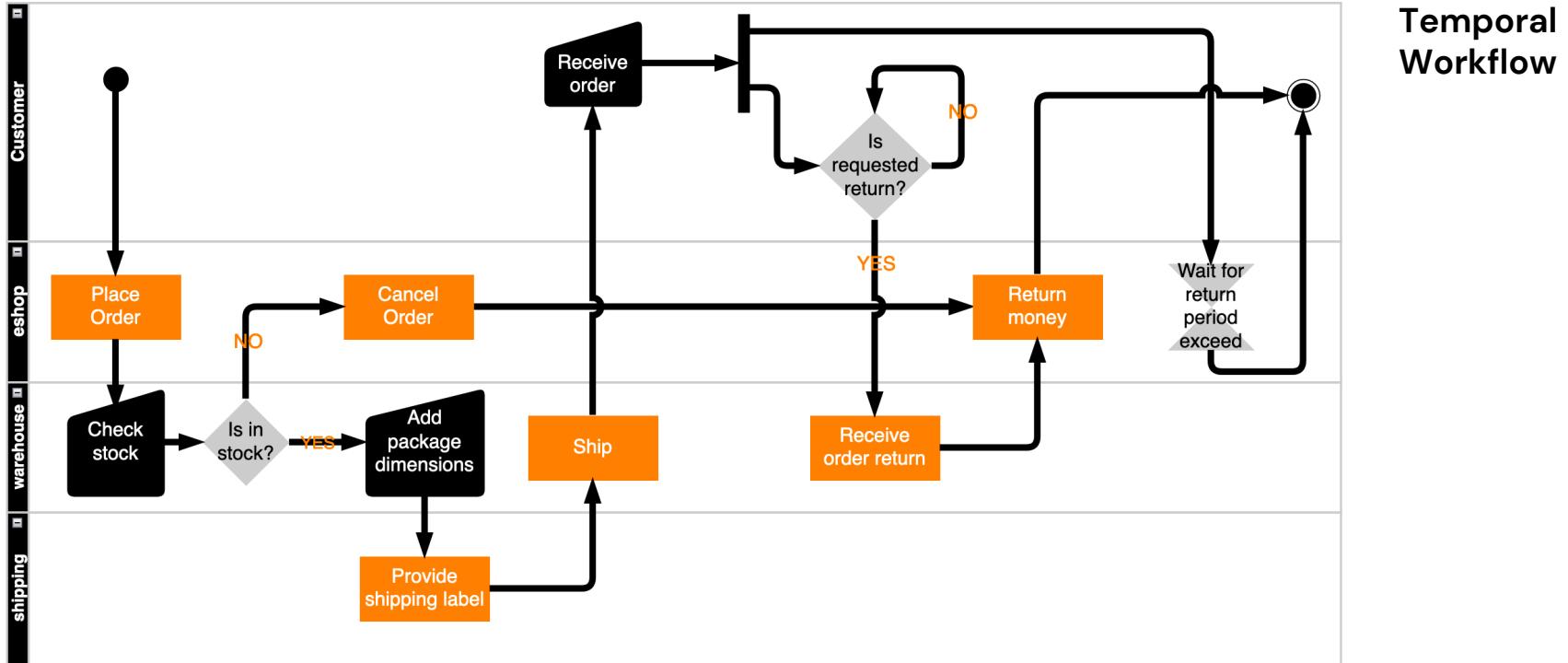


C Possible order processing use case solution with Temporal



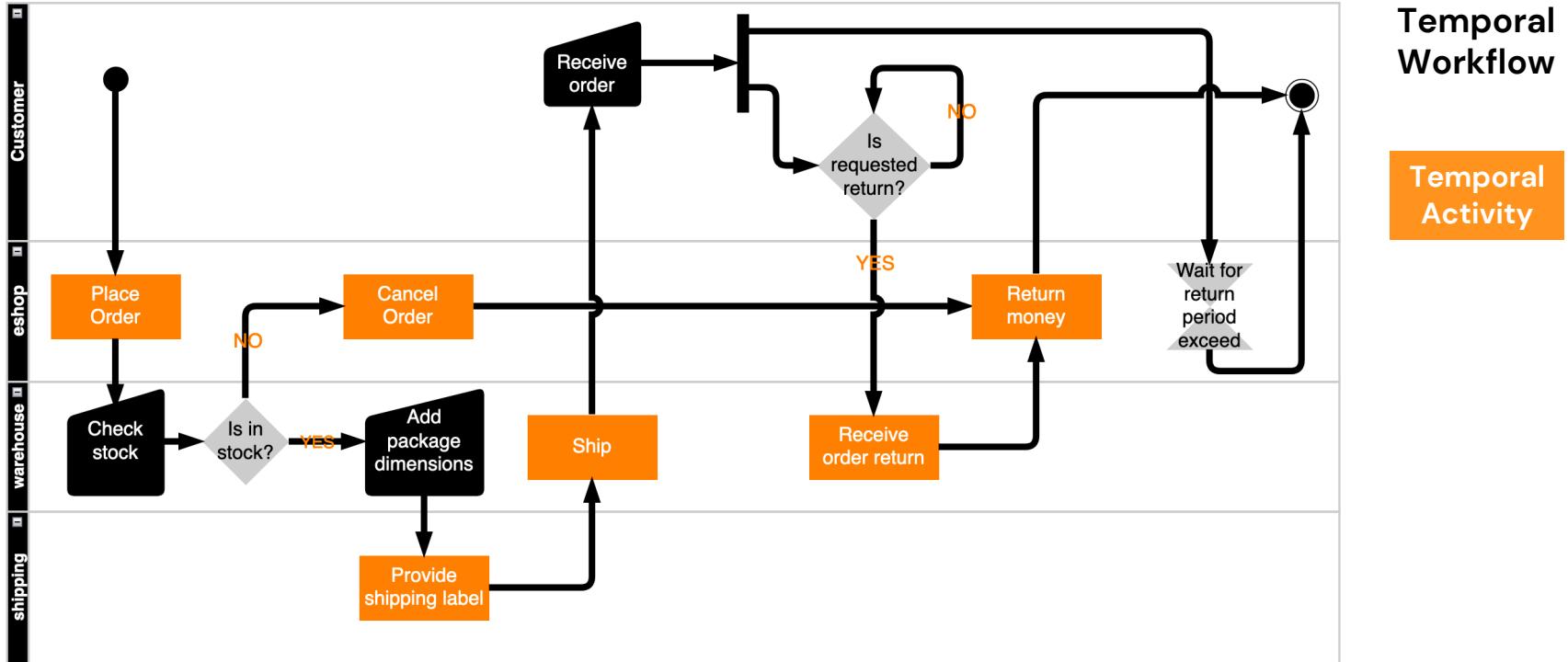


C Possible order processing use case solution with Temporal



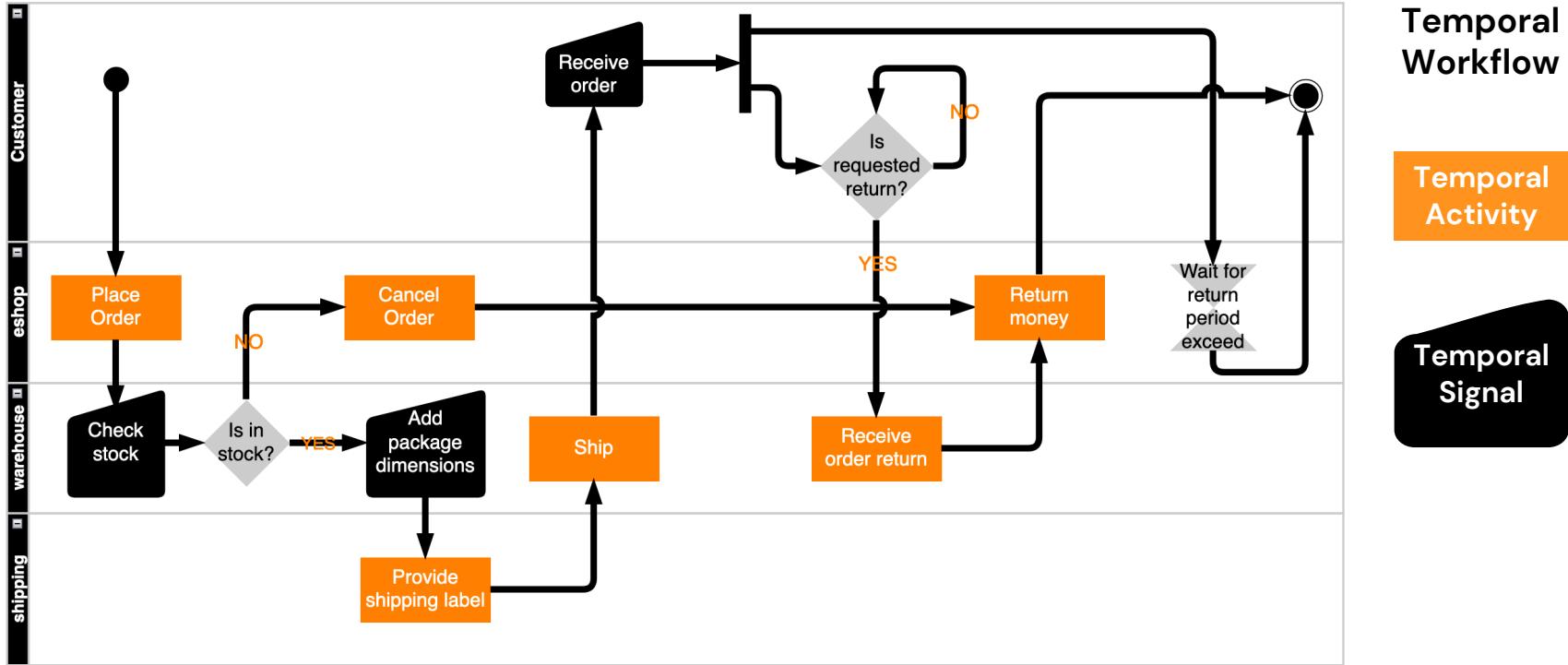


C Possible order processing use case solution with Temporal



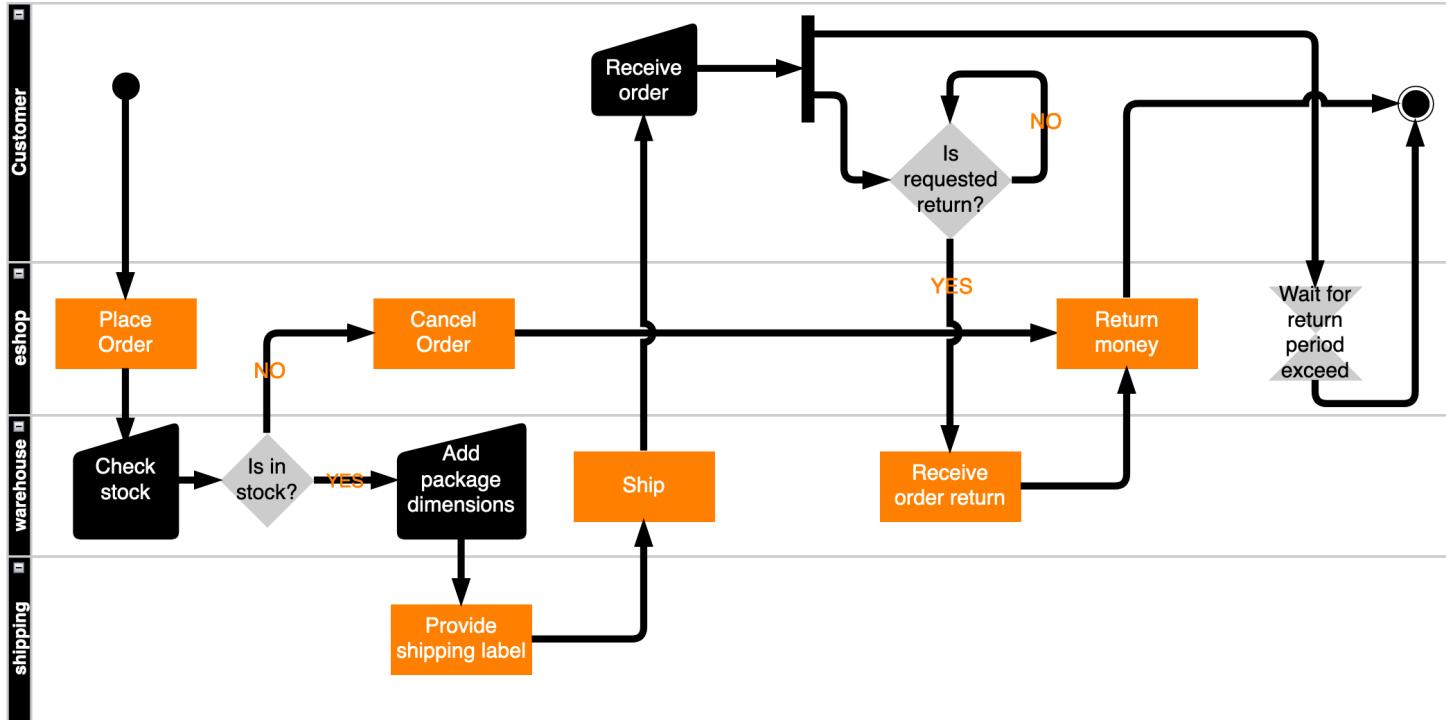


C Possible order processing use case solution with Temporal





C Possible order processing use case solution with Temporal



Temporal Workflow

Temporal Activity

Temporal Signal

Temporal Timeout



D Presentation how it works with Temporal

Let's go to review
the POC
implementation



C Temporal: benefits comparing to “classic” distributed systems



C Temporal: benefits comparing to “classic” distributed systems



Lower complexity
then distributed
systems



C Temporal: benefits comparing to “classic” distributed systems



Lower complexity
then distributed
systems



Business logic
changes do not
force to:

- Change infrastructure
- Add complex workaround



C Temporal: benefits comparing to “classic” distributed systems



Lower complexity than distributed systems



Business logic changes do not force to:

- Change infrastructure
- Add complex workaround



For Order processing use-case it could be:

- Get invoice before providing shipping label
- Ensure transaction (force to implement SAGA)

What should be taken into consideration

- ! PHP workers should be run on RoadRunner (high-performance PHP application server)
- ! Requires familiarity with asynchronous behavior





Q&A





More info

<https://temporal.io/>

<https://temporal.io/get-started-with-php>

<https://youtu.be/t524U9CixZ0>

<https://www.youtube.com/@Temporalio>

<https://github.com/temporalio/samples-php>

