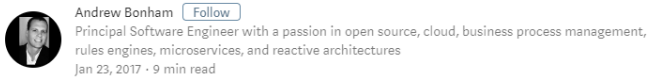
MICROSERVICES ORCHESTRATIONS WITH BPM

Dr. Marigianna Skouradaki

NOVATEC

Microservices…To BPM or not to BPM?



Microservices—When to React Vs. Orchestrate **Source:** https://goo.gl/SVkNkK

Business process management in a "microservices world"

By David Bush October 10, 2016

Source: https://goo.gl/B9y6tH

WEDNESDAY, MARCH 01, 2017

Are MicroServices the Death of BPM and Case Management? **Source:** https://goo.gl/ypBRKZ

Wer Microservices richtig macht, braucht keine Workflow Engine und kein BPMN

01.09.2015 von **Tobias Flohre** 13 Kommentare

Source: https://goo.gl/hTG6nf

Microservices…To BPM or not to BPM?

RED HAT SUMMIT

Source: https://goo.gl/YF951d

Using BPM Suite in a Reactive Architecture with Microservices, Docker, and Amazon ECS

Business process management in a "microservices world"

By David Bush October 10, 2016

Source: https://goo.gl/B9y6tH

Andrew Bonham Follow

Principal Software Engineer with a passion in open source, cloud, business process management, rules engines, microservices, and reactive architectures

Jan 23, 2017 · 9 min read

Microservices—When to React Vs. Orchestrate **Source:** https://goo.gl/SVkNkK

WEDNESDAY, MARCH 01, 2017

Are MicroServices the Death of BPM and Case Management? **Source:** https://goo.gl/ypBRKZ

Wer Microservices richtig macht, braucht keine Workflow Engine und kein BPMN

01.09.2015 von **Tobias Flohre**

13 Kommentare

Source: https://goo.gl/hTG6nf

conductor

Conductor

Conductor is an *orchestration* engine that runs in the cloud.

zeebe by Camunda

Microservice Orchestration Engine

camunda

Microservices and BPM

SWF

…

Agenda

- The principles of microservices
- Challenges in microservices architectures
- Microservices orchestrators
- Real-world practice use case
- Conclusion

VS.

THE PRINCIPLES OF MICROSERVICES

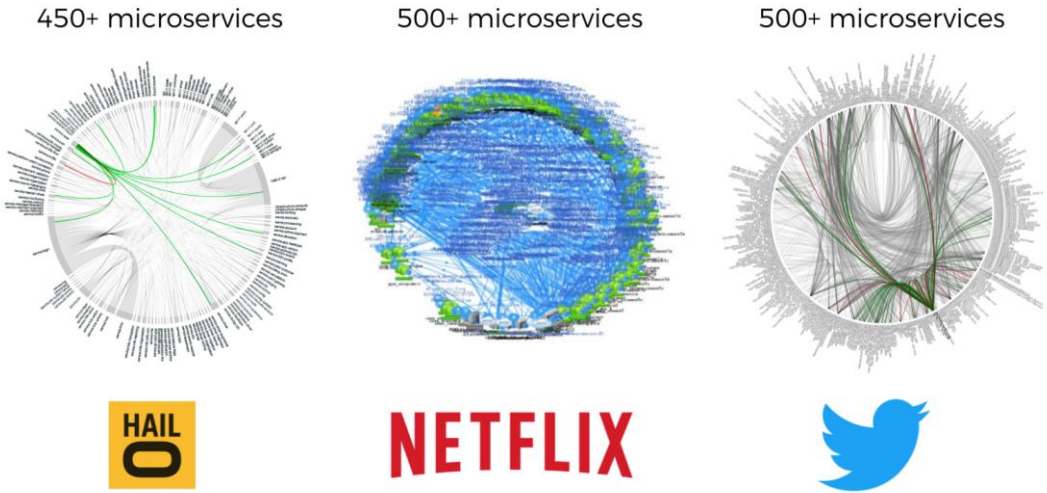
Microservice as a component

1. …do one thing and do it well
2. …organized in “bounded contexts”

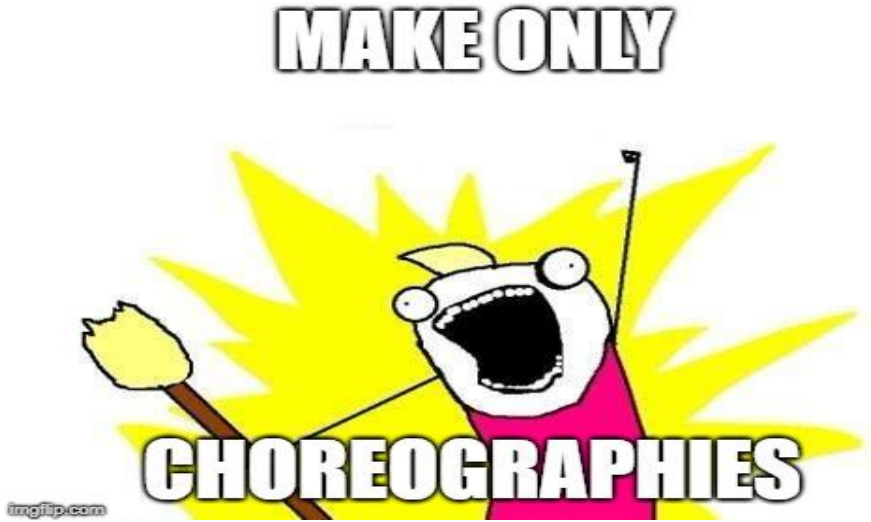
Source: https://www.youtube.com/watch?v=9gLrCPVrXo4

Microservices as a system (Lewis J. & Fowler M., 2014)

1. Smart endpoints – Dumb pipes
2. Decentralized governance
3. Decentralized data management
4. Infrastructure automation
5. Design for failure
6. Evolutionary design



Source:http://heidloff.net/article/introduction-reactive-microservices



- ✓ Systems are loosely-coupled
- ✓ No central-coordinator
- ✓ Bounded contexts are clear
- ✓ Microservices encapsulate own logic and data
- ✓ Event channels act as dumb pipes

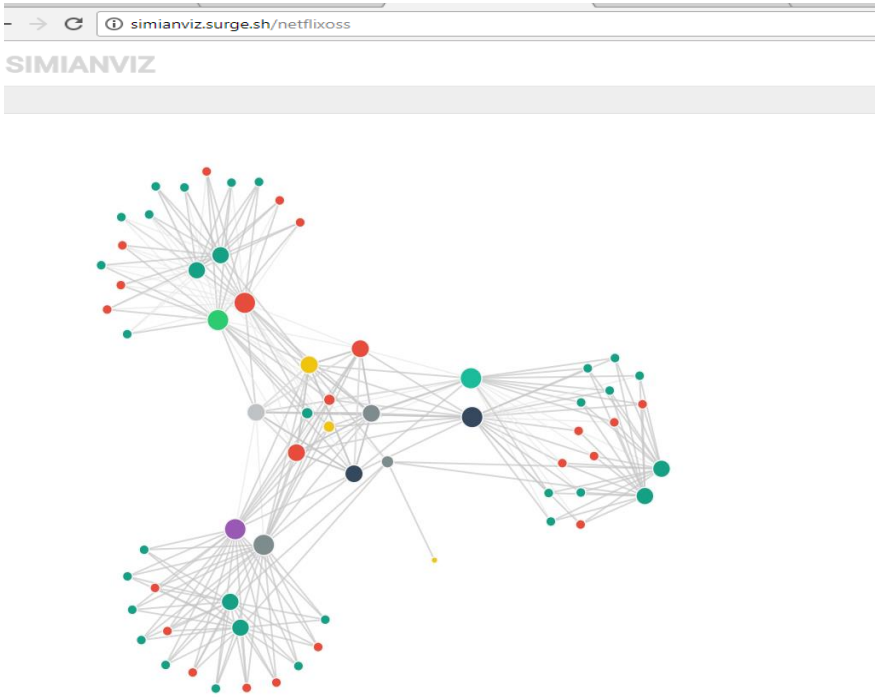
„With peer-to-peer choreography,
we found it was harder
to scale with growing needs
and complexities“

Viren Baraiya *& Vikram Singh*

NETFLIX

CHALLENGES IN MICROSERVICES ARCHITECTURES

Flows are hidden



It is difficult to clarify which microservice interacts with which.

There is no explicit end-to-end processs.

Screenshot of SimianViz demo for complex Netflix visualization of microservices

“Design for Failure” – Failover logic

Source: http://www.elblender.com/wordpress/wp-content/uploads/2016/12/Microservice-Resilient.png

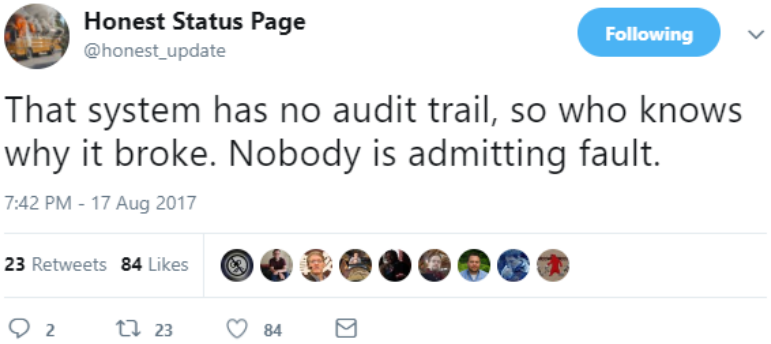
- Stateful retry in asynchronous manner
- Self-healing
- Failover Caching
- Retry Logic (retry with timeouts, retry after failure)
- Error Handling in Transactions (SAGA Pattern)

Transparency of status

How much is done until X completes?

Source: https://goo.gl/p3Ujws

Monitoring

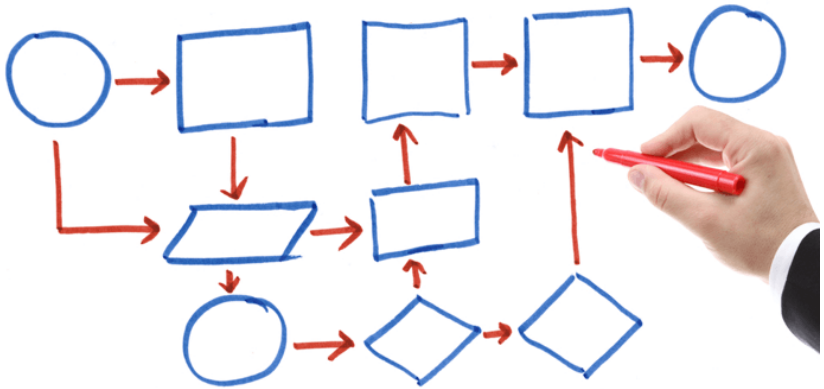


Have one place with gathered information about all running processes and collaborations

Managing a business process

Be able to manage a process –
i.e., run, restart and stop per need.

BPM solves it all..

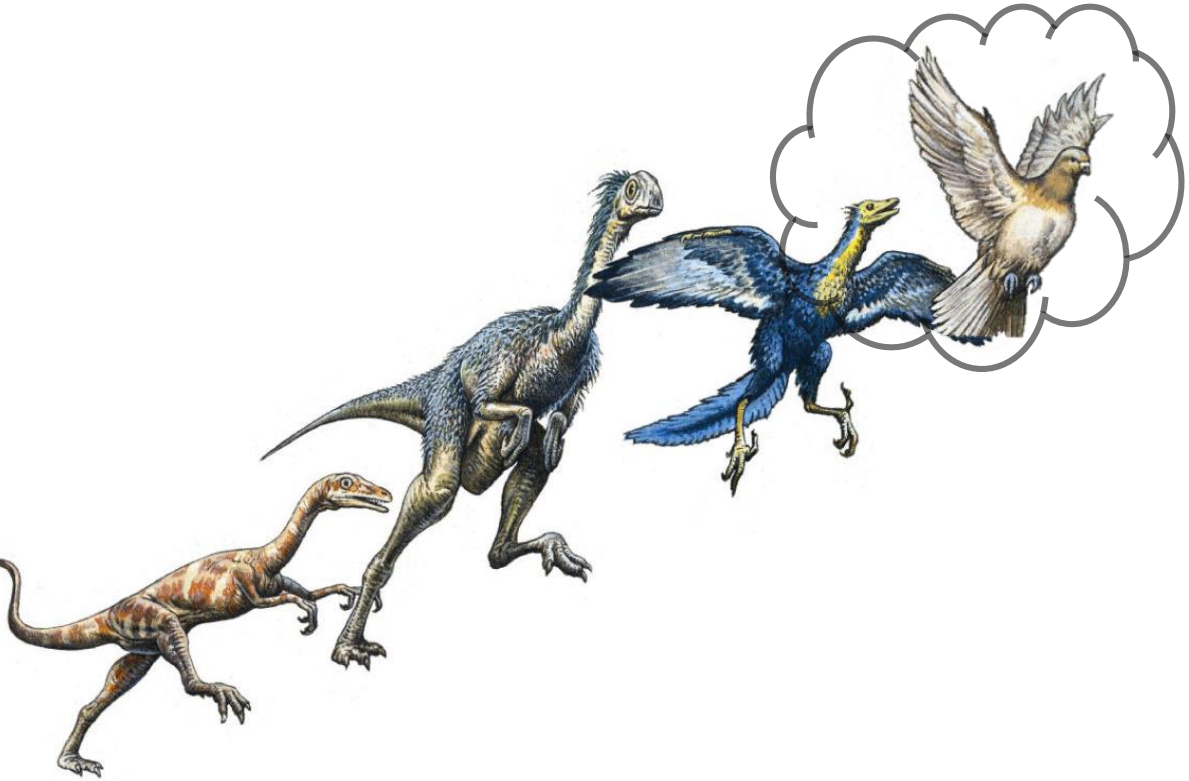


Source: https://goo.gl/EGhxrb

By construct…

- …workflows provide a visual overview of a business process
- …workflow engines provide management of a business process
- …workflow engines provide monitoring, audit logs and history

Process engines have evolved



- Embeddable
- Lightweight
- Cloud-ready
- Cloud-licensing models

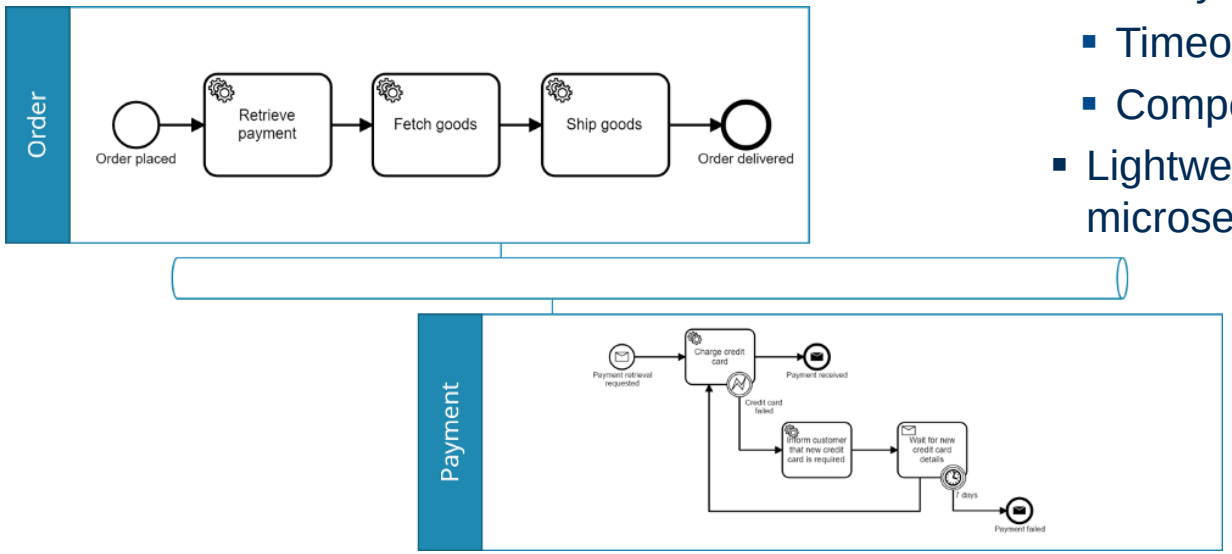
→ Can participate in microservice-friendly architectures

MICROSERVICES ORCHESTRATORS

“Classic” BPMN 2.0 process engines

camunda

Source: https://blog.bernd-ruecker.com/avoiding-the-bpm-monolith-when-using-bounded-contexts-d86be6308d8



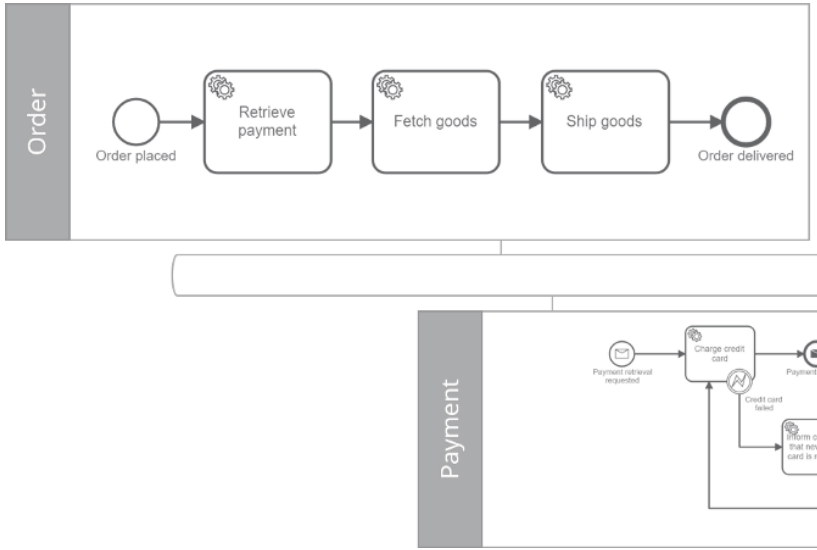
- BPMN 2.0 based process engine
  - Retry
  - Timeouts
  - Compensation
- Lightweight – can run embeddable in the microservice

“Classic” BPMN 2.0 process engines

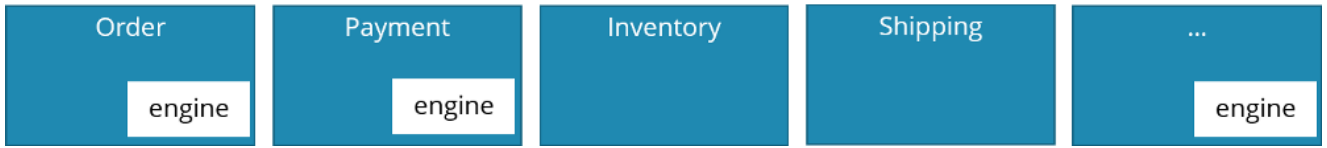
camunda

- BPMN 2.0 based process engine
  - Retry
  - Timeouts
  - Compensation
- Lightweight – can run embeddable in the microservice

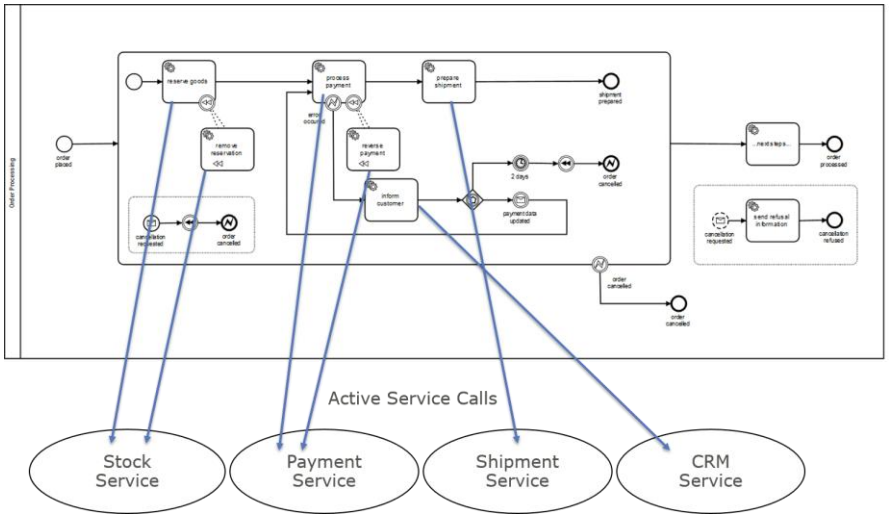
Source: https://blog.bernd-ruecker.com/avoiding-the-bpm-monolith-when-using-bounded-contexts-d86be6308d8



zeebe by Camunda



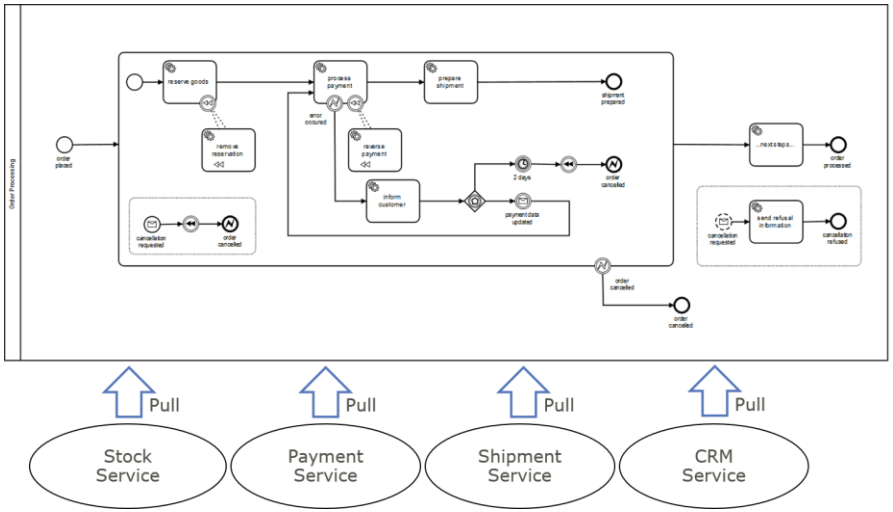
External tasks pattern



PUSH

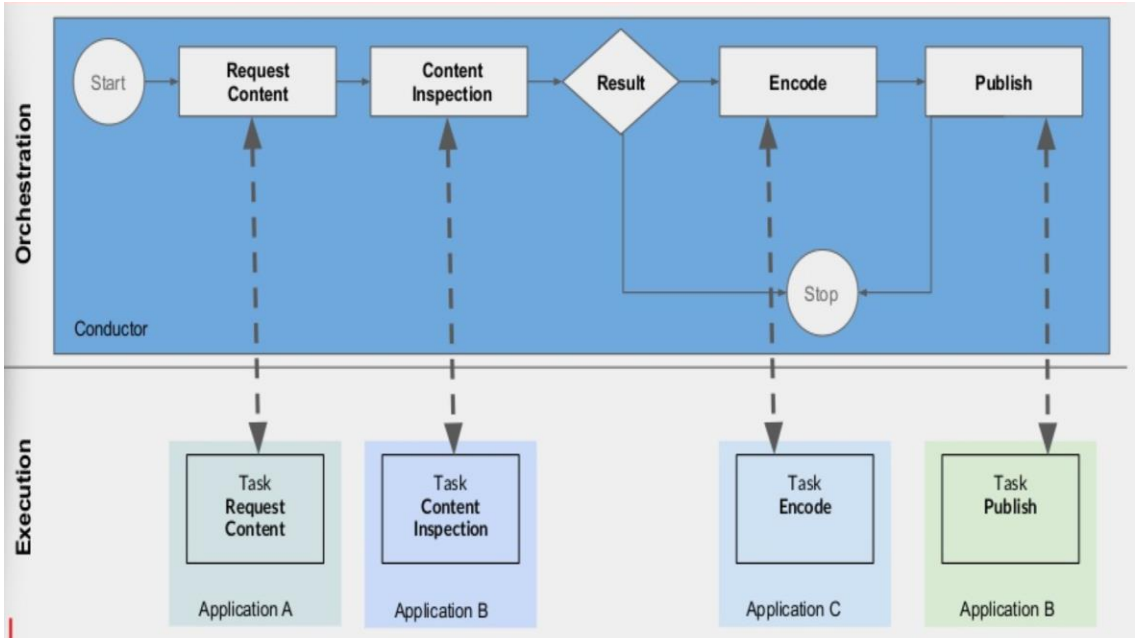
Source: http://www.bpm-guide.de/wp-content/uploads/2015/04/service-call-pull.png

Source: http://www.bpm-guide.de/wp-content/uploads/2015/04/service-call-pull.png



PULL

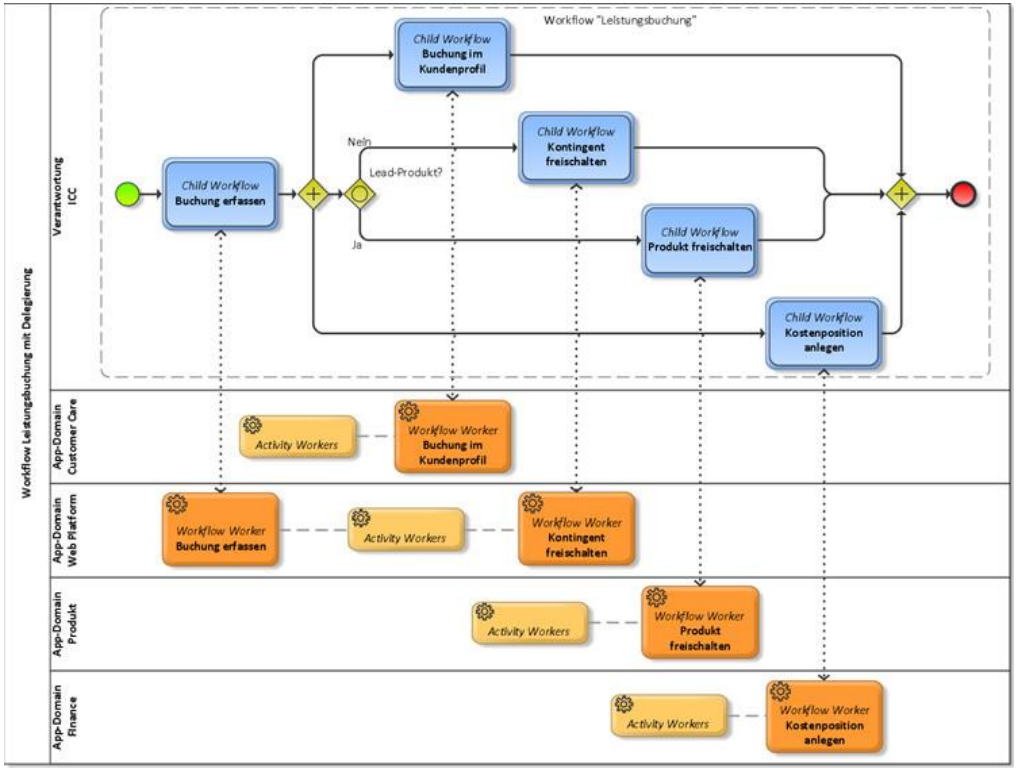
Netflix conductor



- Visualization of a process
- JSON-based DSL Language to define blueprint
- Management of a process
- Traceability
- Retries – Timeouts are allowed in task definitions
- Error/Failure handling

Source: https://image.slidesharecdn.com/netflixconductor-170210192244/95/netflix-conductor-12-638.jpg?cb=1487089280

Amazon Simple Workflows (SWF)

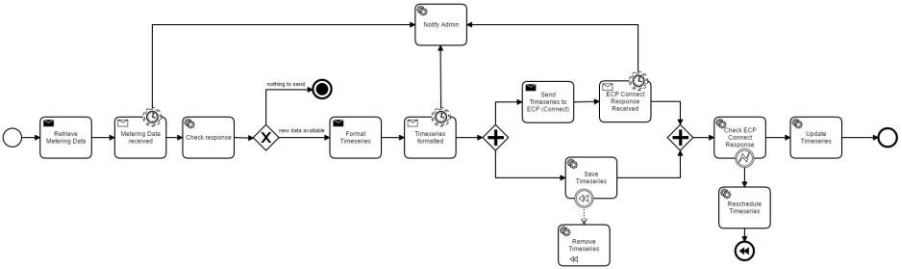


Source: https://media.licdn.com/dms/image/C4E12AQE-x7HbP_uDtw/article-inline_image-shrink_1000_1488/0?e=2127081600&v=beta&t=Bad-UxUAkeobq42NyOdVMlGGNm-X7su_q43jLQDD_DU

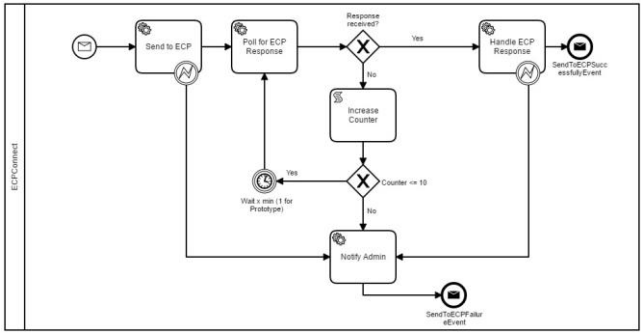
- Decider is a component were we delegate the logic.
- Microworker / decider (is a component where we delegate the business logic) → Microworkers have microworkers
- Workflow engine is only a working horse when it comes to message delivery, state management and error handling.
- Cause a nested sequence of workflow calls to “abstract” logic

REAL-WORLD PRACTICE USE CASE

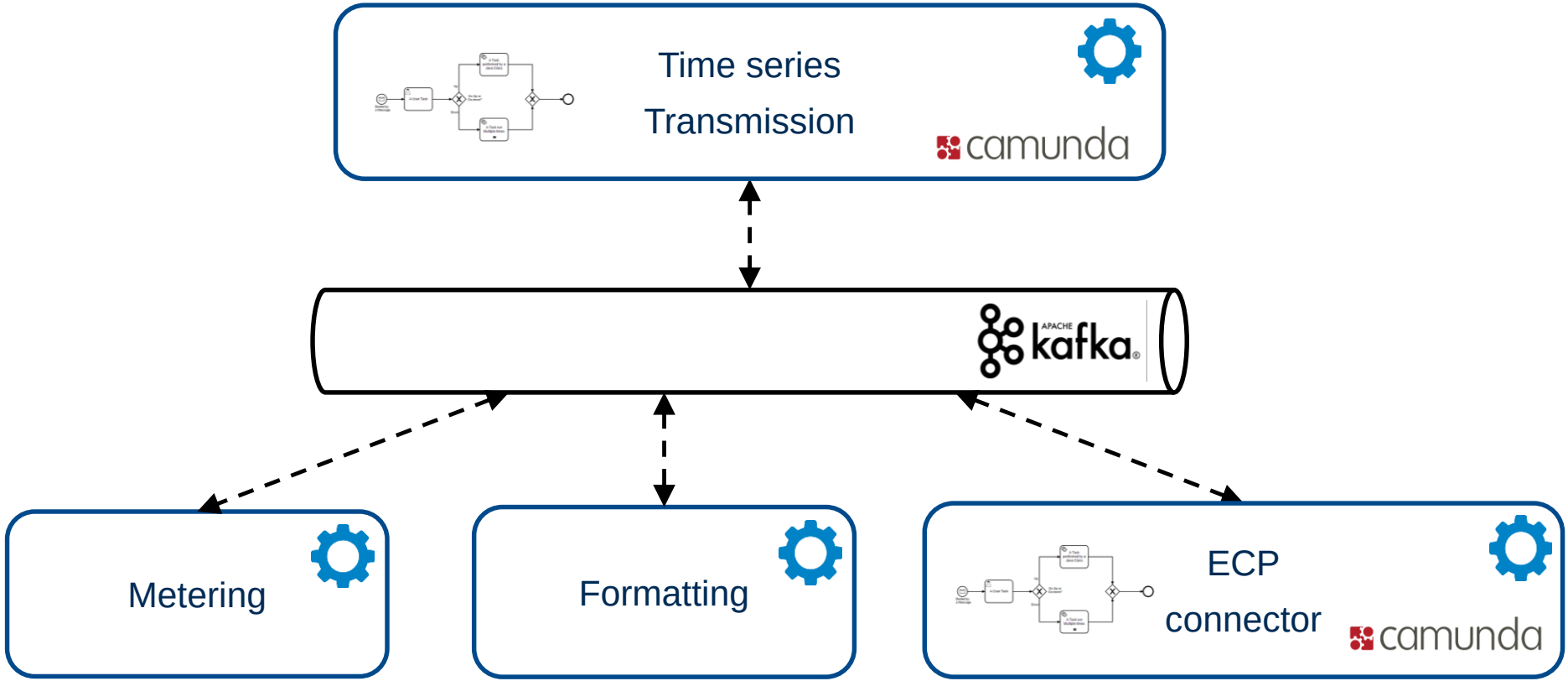
Time Series Process



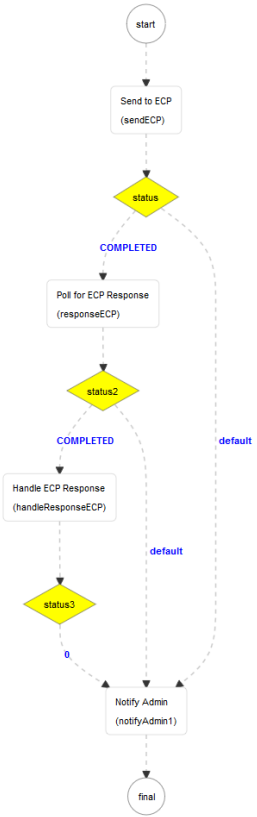
ECP Connect Process



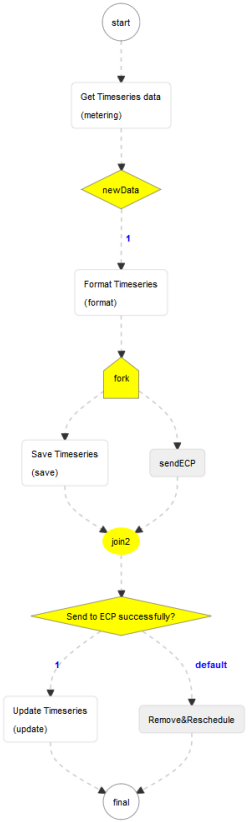
Prototypical implementation with Camunda



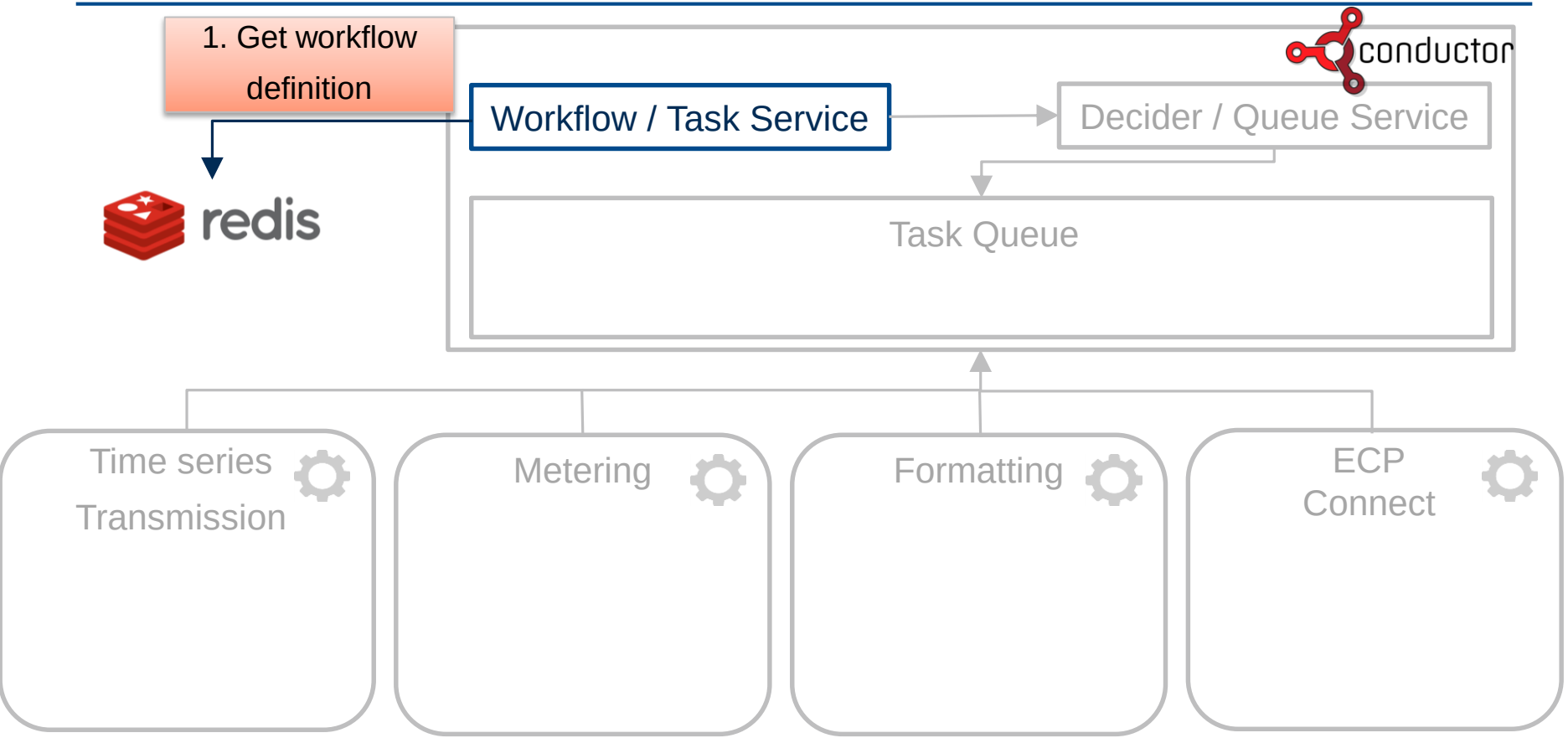
Time Series Process



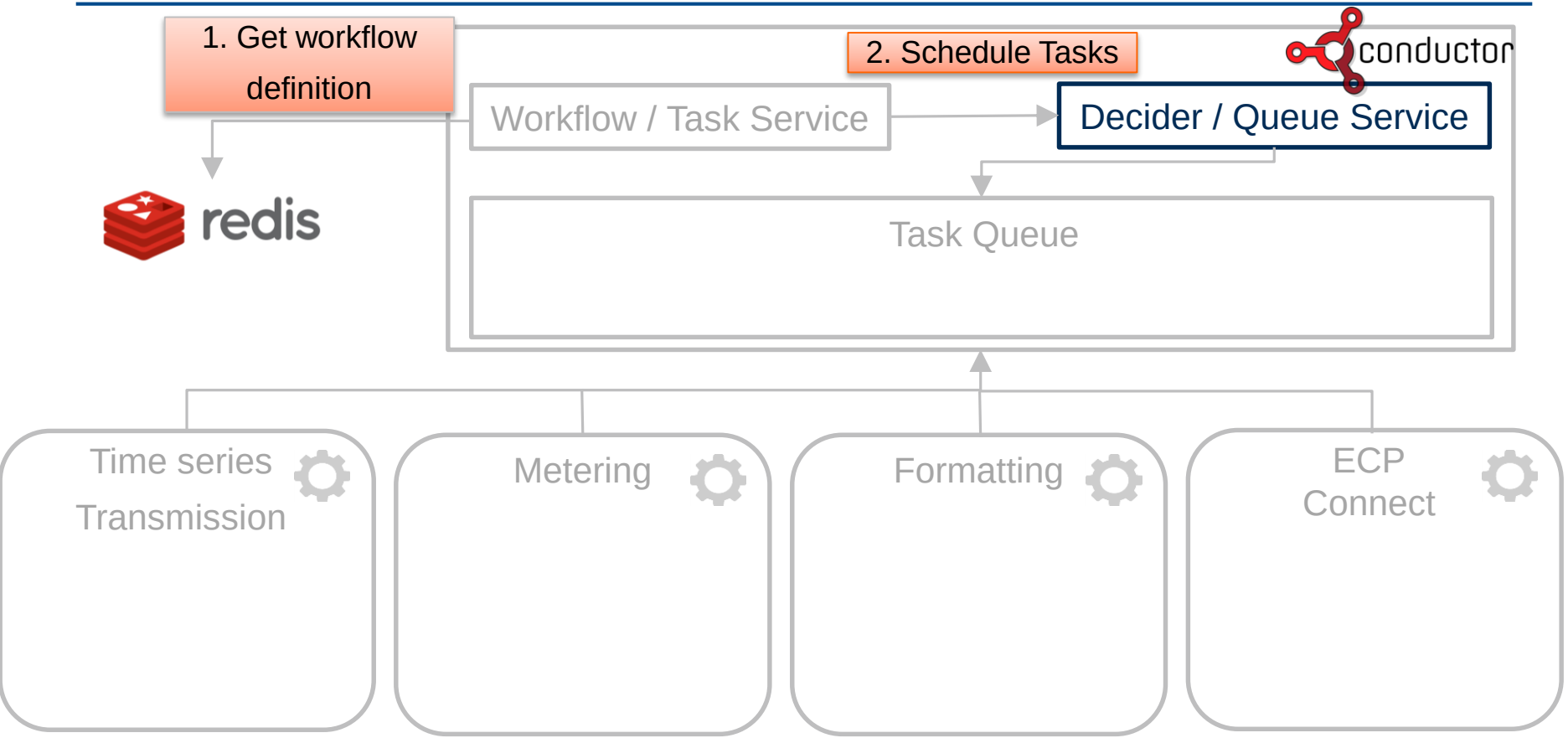
ECP Connect Process



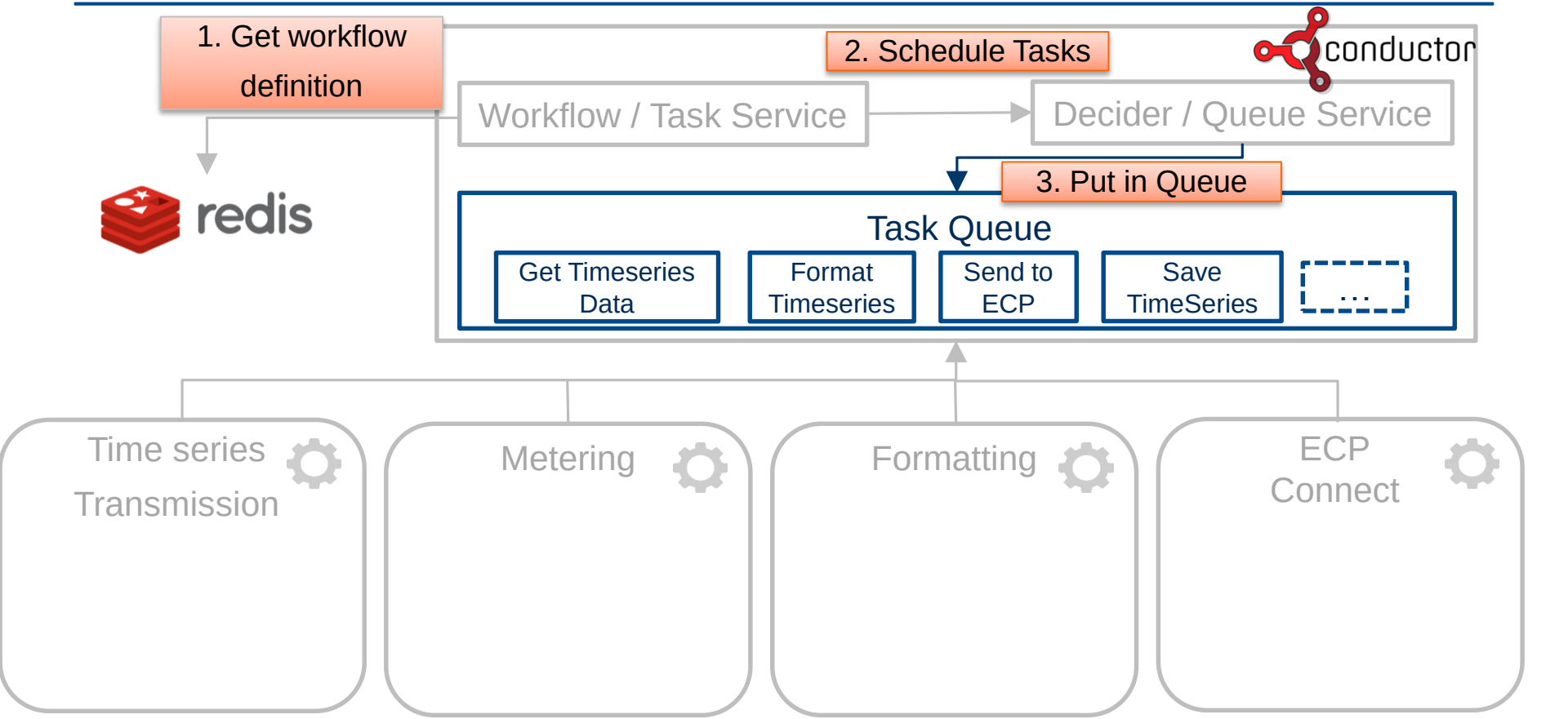
Prototypical Implementation with Conductor



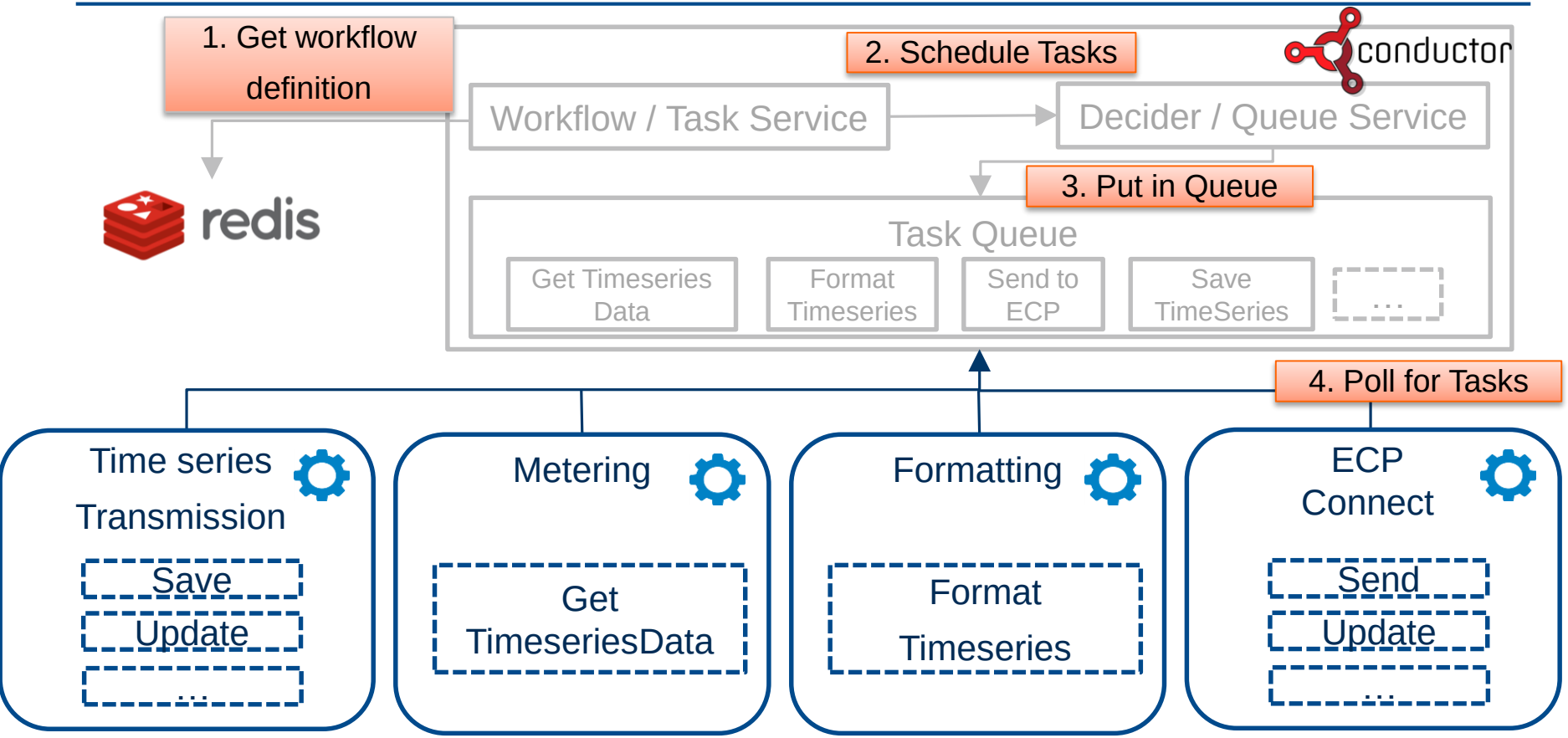
Prototypical Implementation with Conductor



Prototypical Implementation with Conductor



Prototypical Implementation with Conductor



Observations (Delegate, Delegate, Delegate!)

- Delegation is key
- Simple flows instead of complex business processes
- Pull Tasks instead of pushing
- “Smart” workers and “dumb” engine
- New terminology commonly used in most tools
  - Flows
  - Decider
  - Worker
  - Task
  - …

Source: https://goo.gl/QYnuf7

Conclusions & Future Work

- Microservices have also changed the BPM landscape
- The same fundamental reasons have led the “market leaders” to develop orchestration solutions
- Microservice orchestrators provide a **unified but not unique** way to visualize, monitor and manage a distributed process
- Each architectural choice comes with trade-offs and should be chosen per use case


@novatecgmbh


novatec-gmbh.de/karriere/