

Titiruck Nuntapramote

Titiruck.nuntapramote@rwth-Aachen.de

# Adapting Regression Test Optimization for Continuous Delivery

Final Talk | Master Thesis | 31.10.2018

Supervisor:

Dipl.-Inform. Andreas Steffens

Christian Plewnia, M.Sc.

## Motivation

---

- In continuous delivery, we aim to automate as much as possible
  - That should not include **just regression testing but RTO as well!**
- Why do we want to integrate RTO into the delivery process?
  - So far, RTO in CD is still an under-developed territory
  - Not even part of **CD maturity model** for test & verification!
  - No **generic RTO tools** that offer integration into current CD tools
- Make the adoption of automatic RTO in the delivery process as easy as possible!

**Q1.** How to design the architecture of RTO adaptation such that it fits in the architecture of the delivery system?

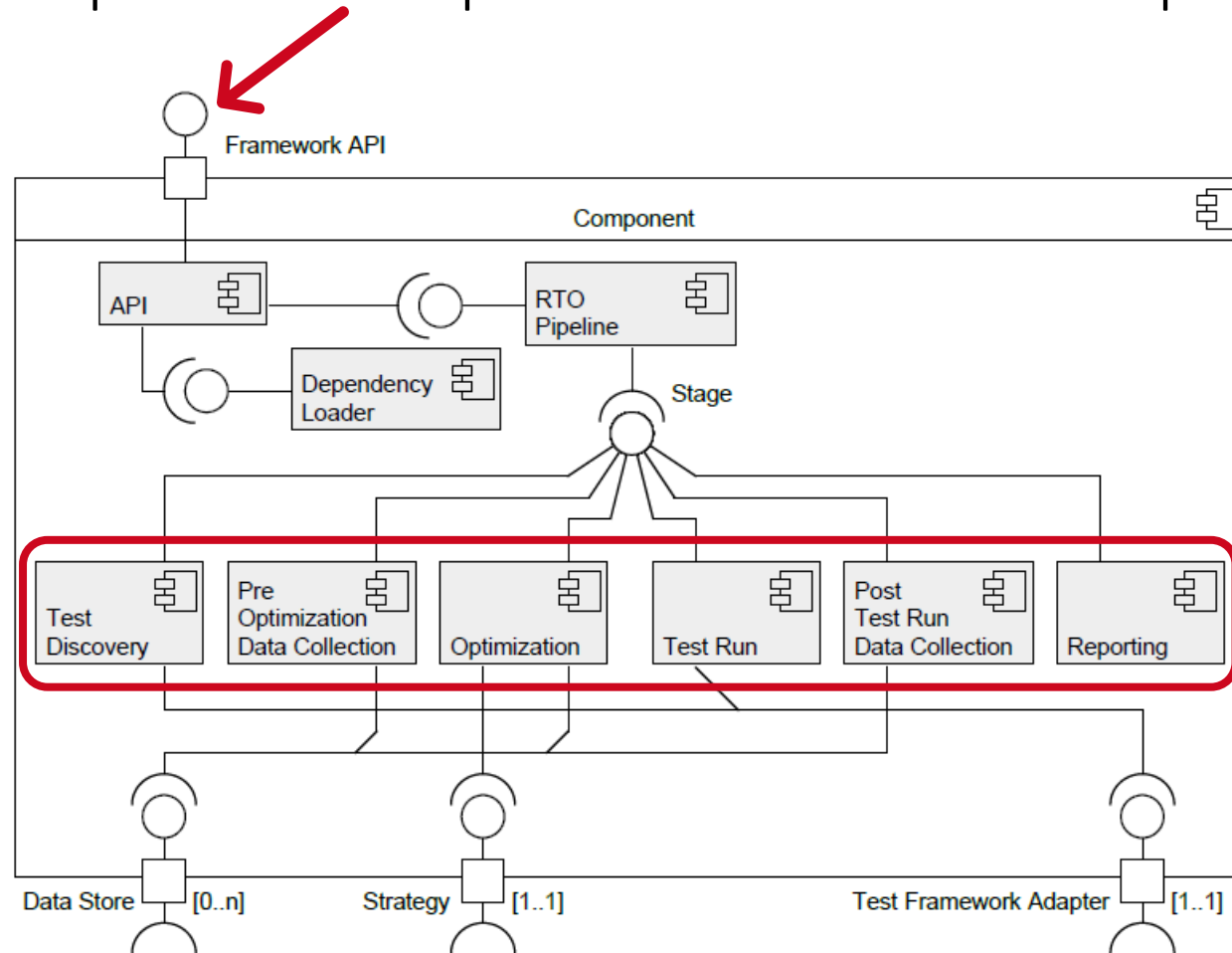
- Decomposition of monolithic RTO into microservices

**Q2.** Does the architecture of the delivery system need to be changed or extended to support the RTO adaptation? And what is the impact of the RTO adaptation to the overall architecture?

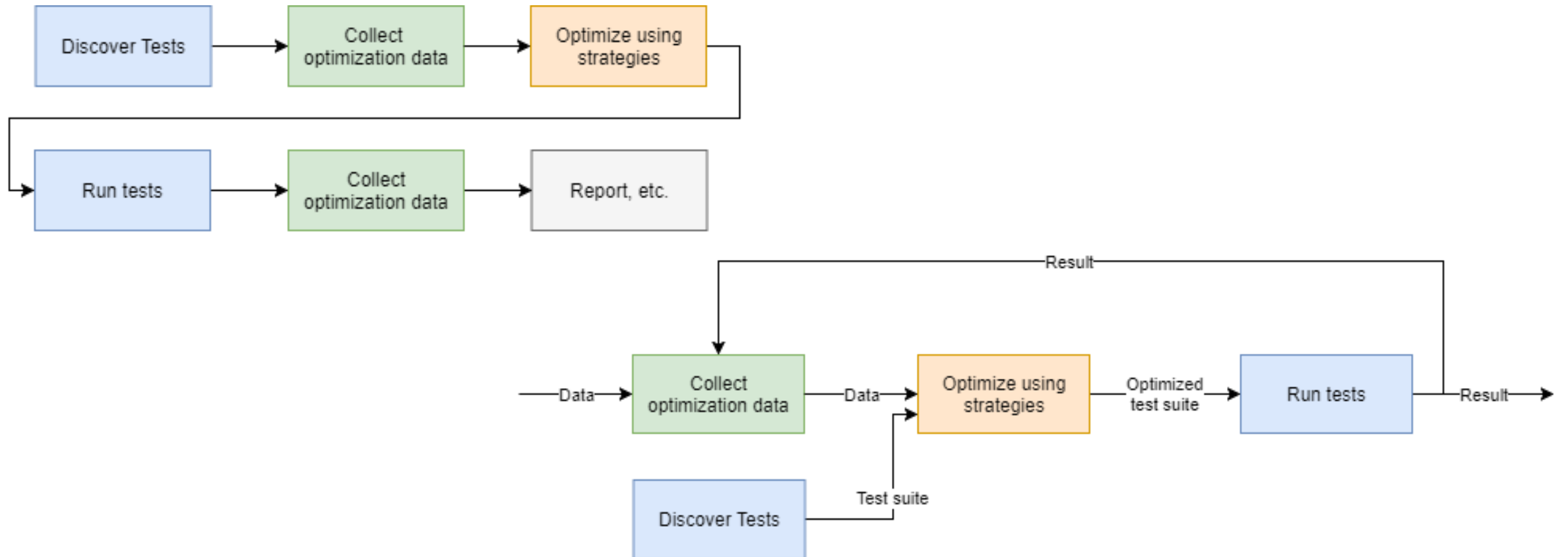
**Q3.** What are the properties of the RTO adaptation? Does the microservice architecture have any advantages or disadvantages over the monolithic architecture?

## Lazzer: Monolithic RTO prototype

- Basis for the conceptual decomposition of the RTO adaptation

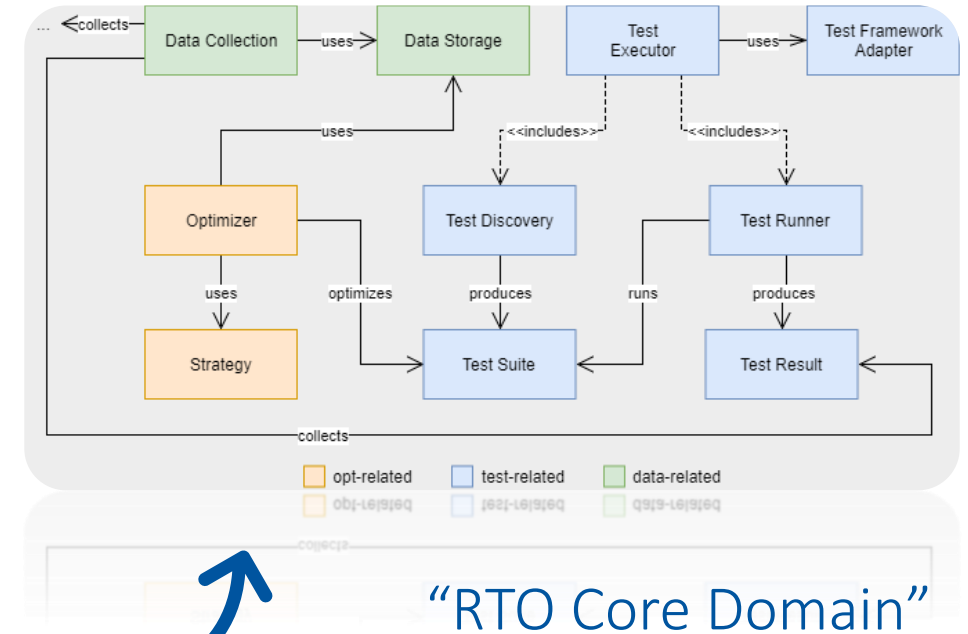
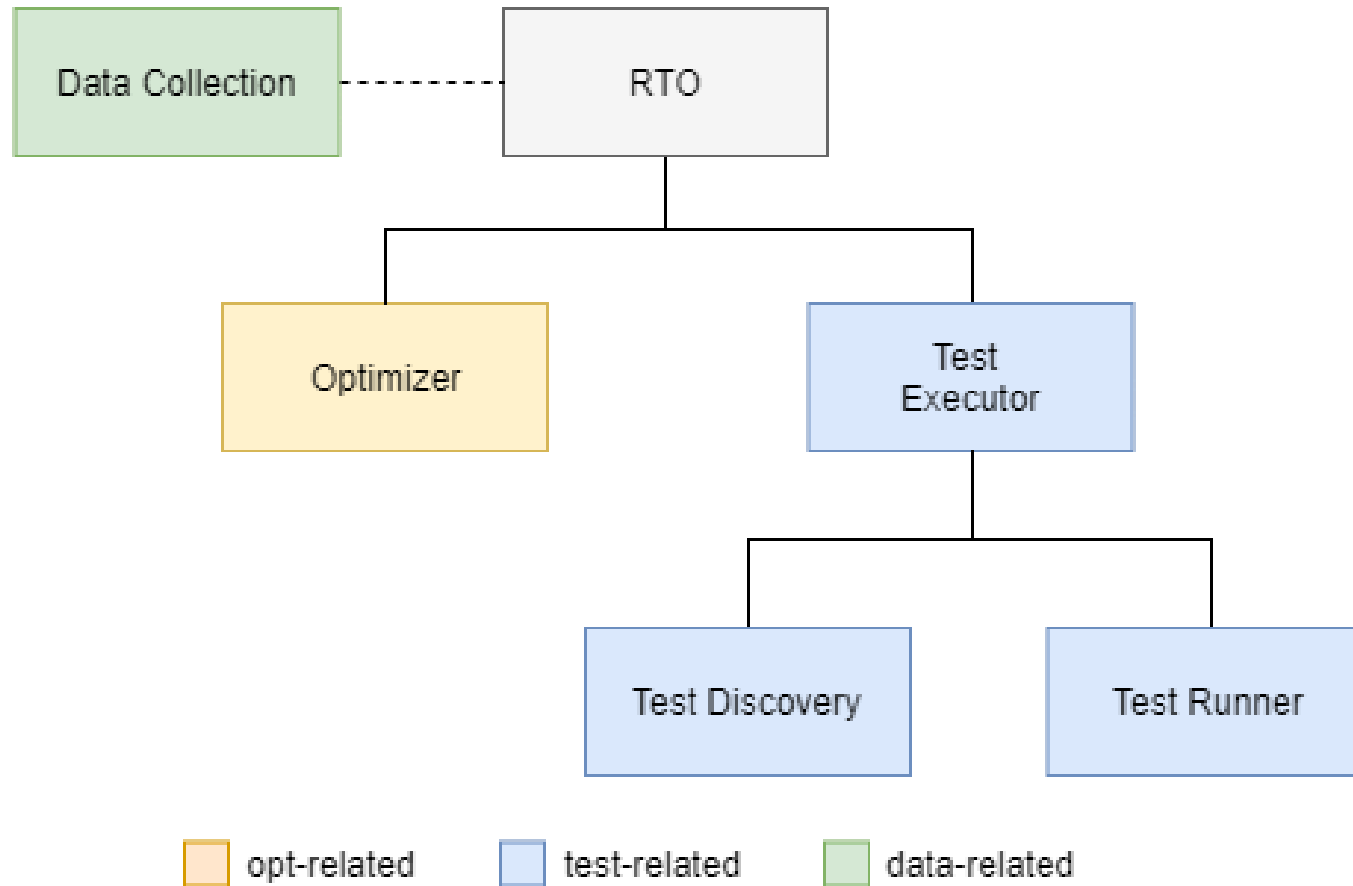


# RTO Pipeline



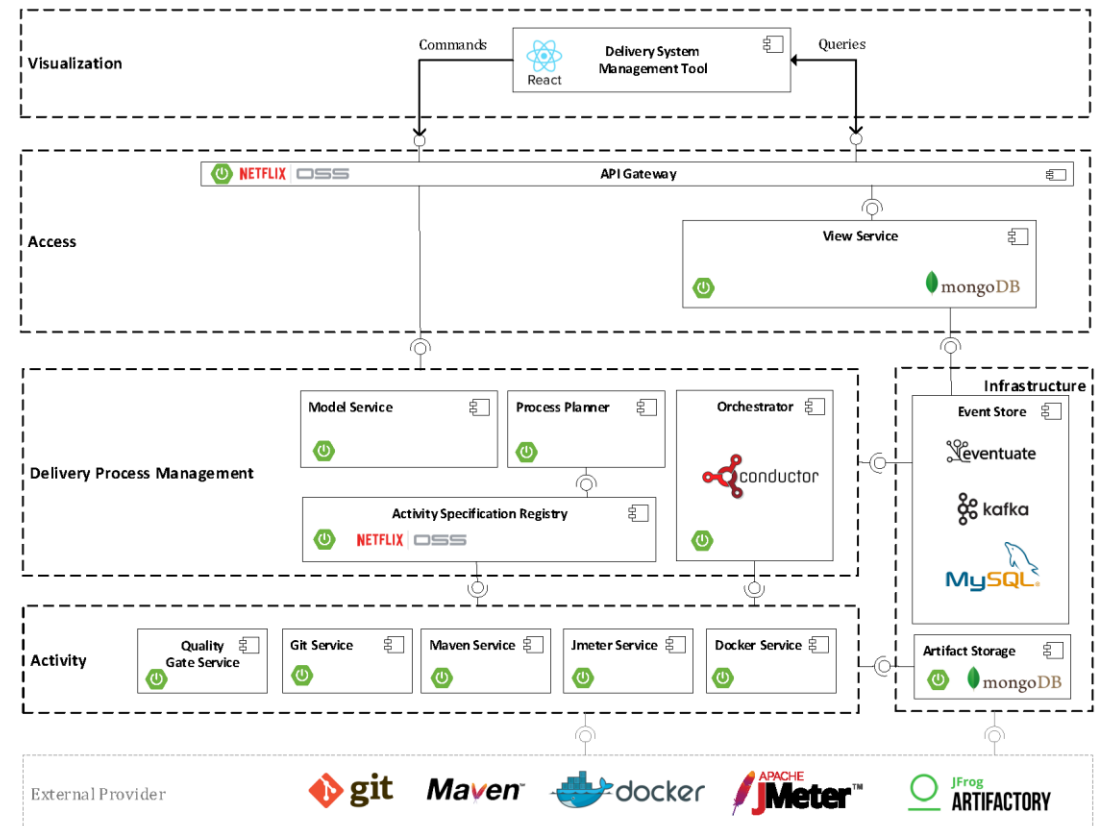
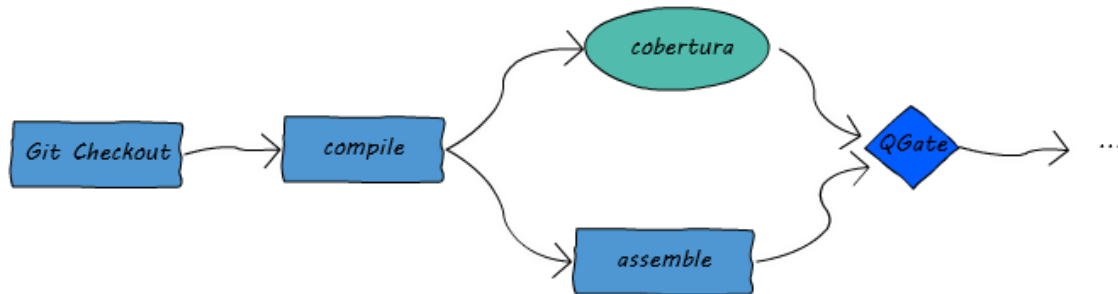
- Divided into 4 areas: ■ opt-related ■ test-related ■ data-related ■ others

# RTO Conceptual Decomposition



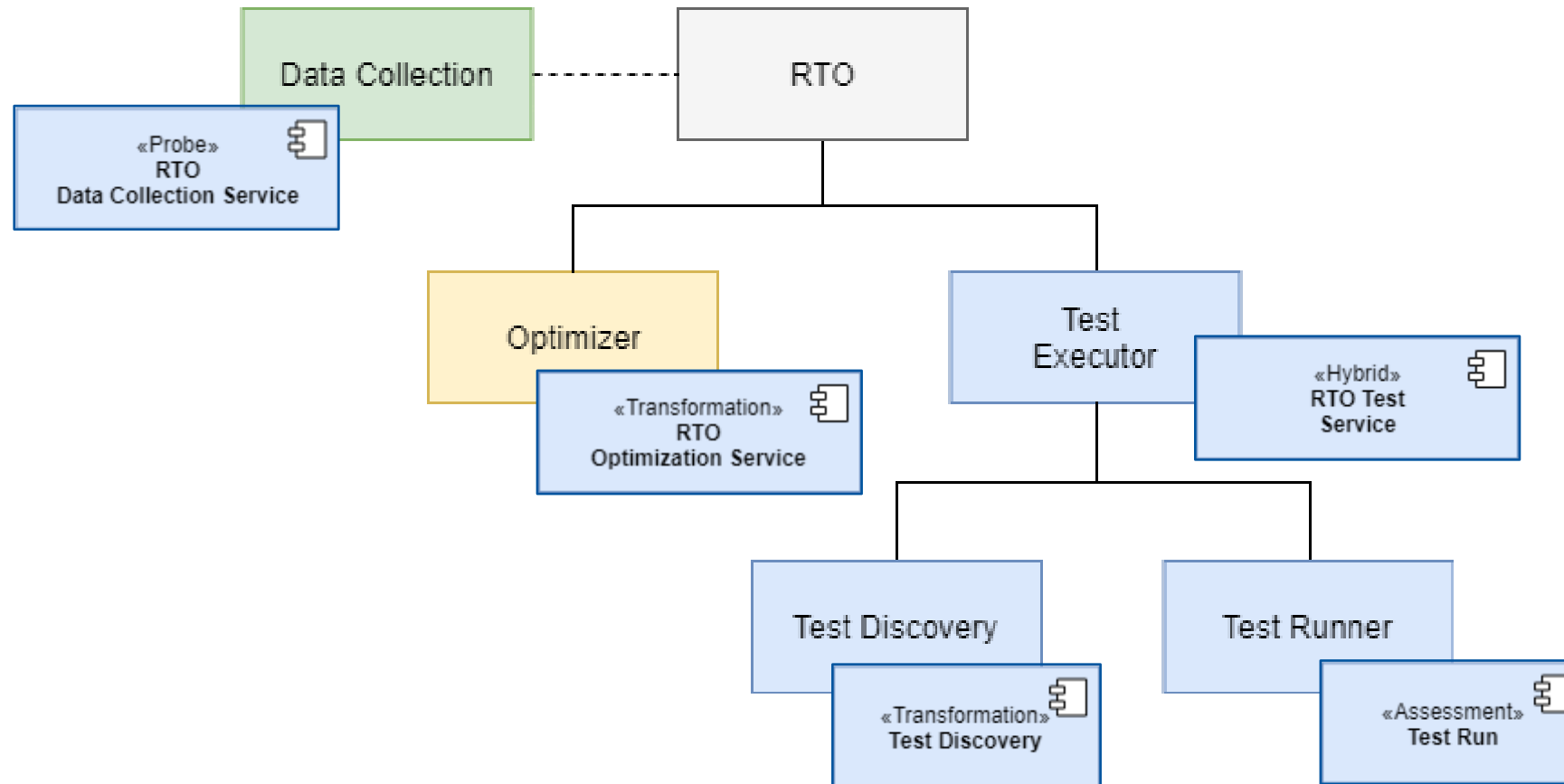
# Jarvis: Self-Organizing Delivery Systems

- Multiple layers
- **Microservices**: modularity, decoupling
- Activity abstraction
- Delivery model made up of activities



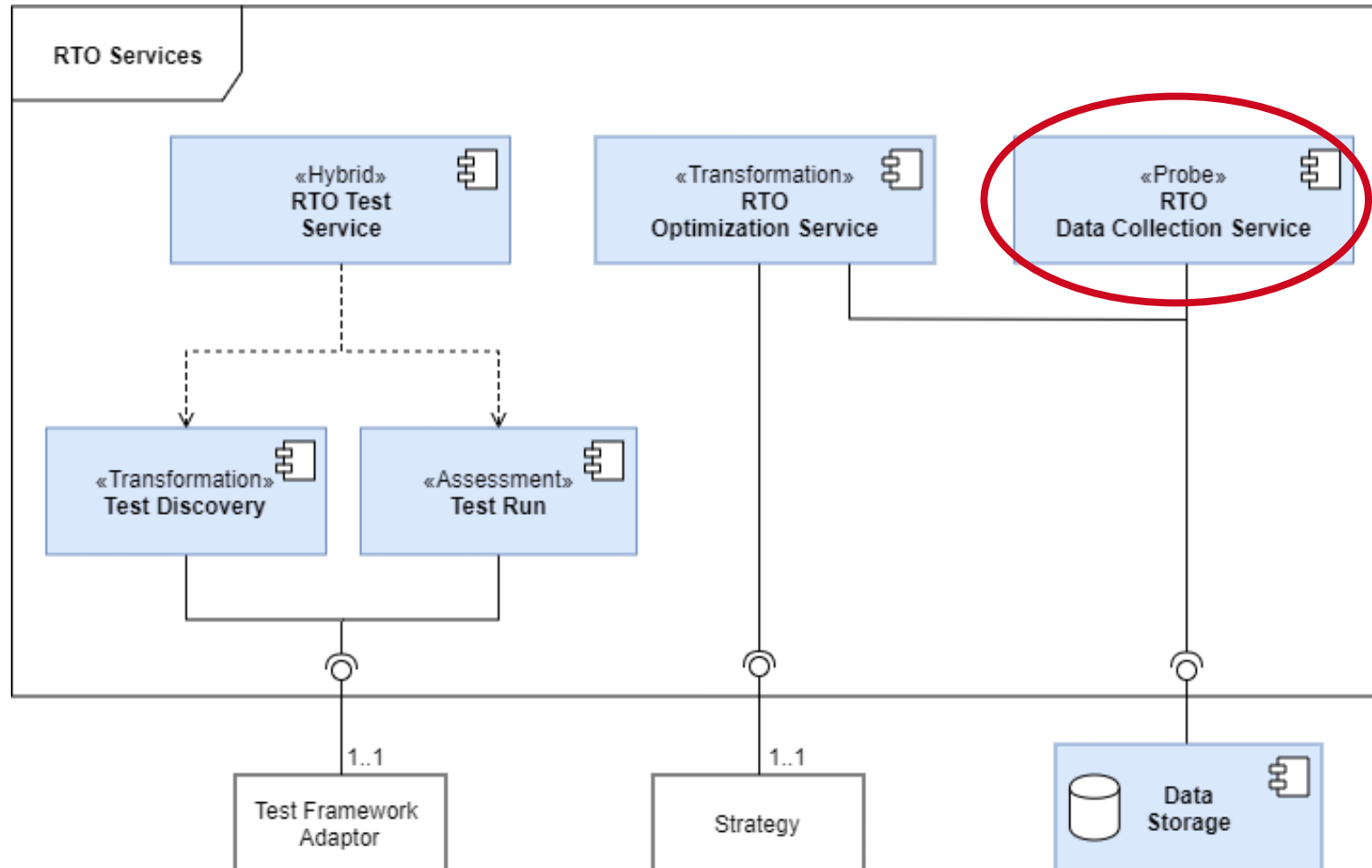
## RTO as Microservices

- Constrained by Jarvis to be realized as microservices



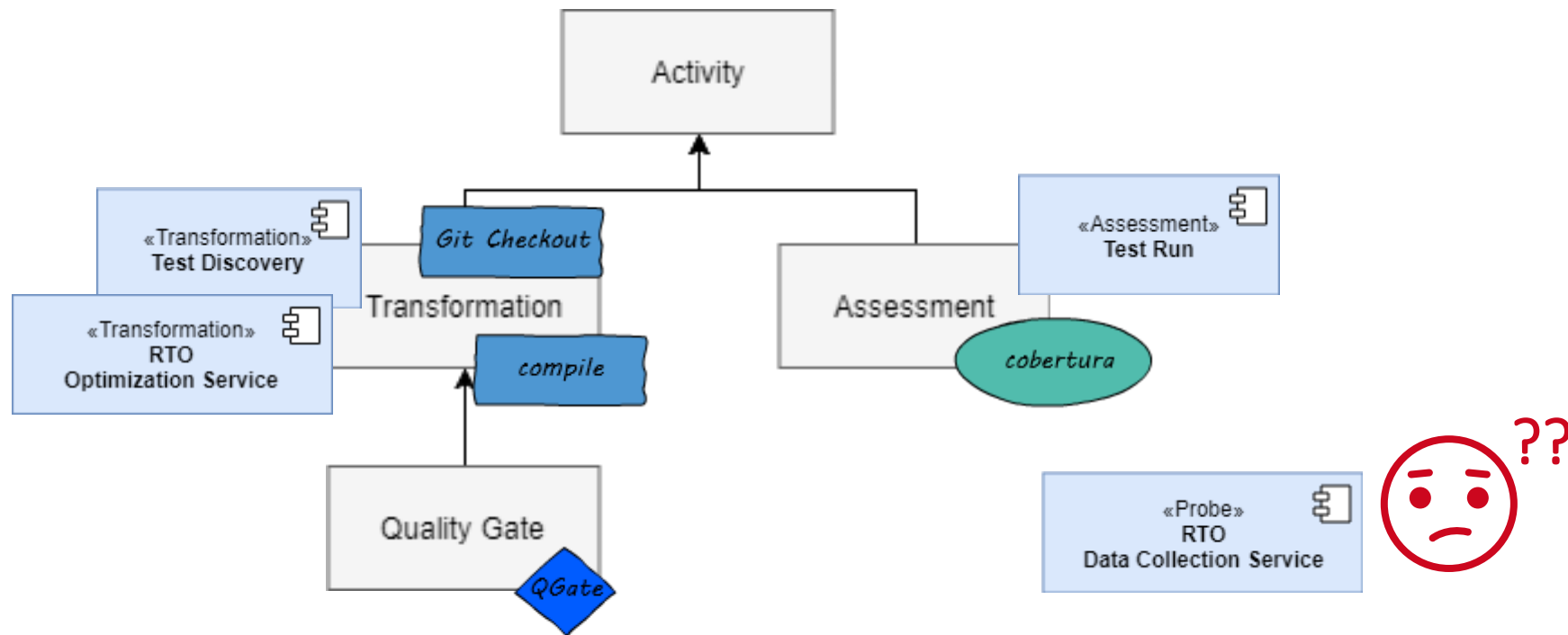


# RTO Adaptation Architecture



## Integration of RTO into Jarvis

- In Jarvis, new technology can be integrated as an activity
  - Straightforwardly we can implement RTO services as activities



## What is the Problem?

---

- But...what about data collection service?
  - Does not transform!
  - Does not assess!
  - Does not need to be evaluated by quality gate, unlike assessment
  - Basically, does not contain any logic other than probing and storing data into data source

## What is the Problem? (2)

---

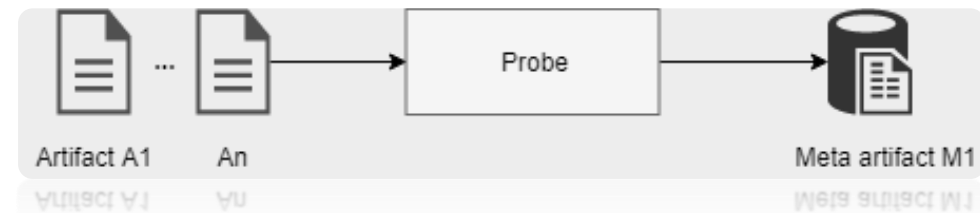
- RTO pipeline run is not atomic
  - We need data collected from the previous runs
  - We need to distinguish each run when probing/storing data
- **But... the delivery process run is atomic!**
  - No access to other runs
  - No concept of activity being aware of its environment (delivery process)

## Probe: Definition

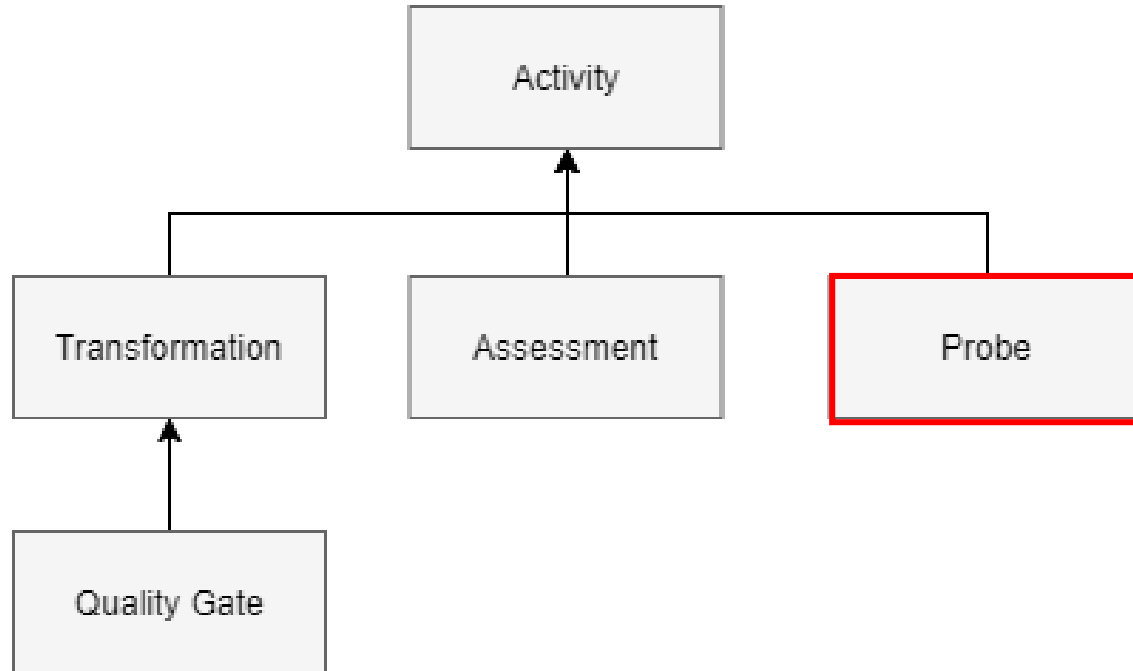
---

“**Probe** is an activity that takes a number of artifacts , probes for a partial state of data from the said artifacts, and store them in some data storage.”

- Probed data is referred to as “**Meta artifact**”
- Collect data
  1. About the input artifact(s)
  2. About the delivery process
  3. About the delivery system

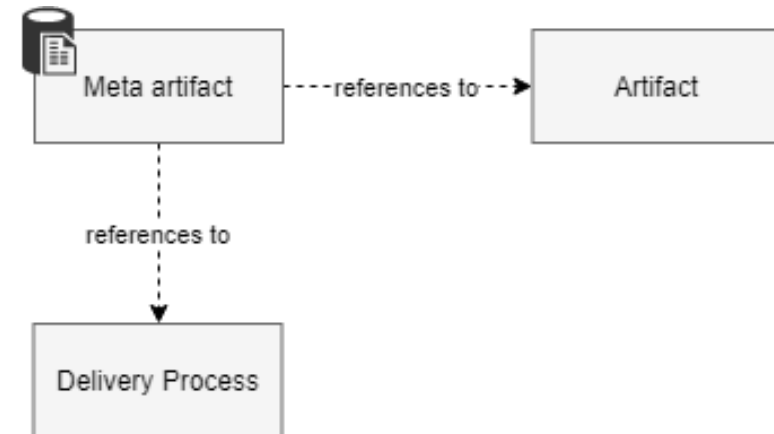


# Probe: Activity Classification and Context Awareness



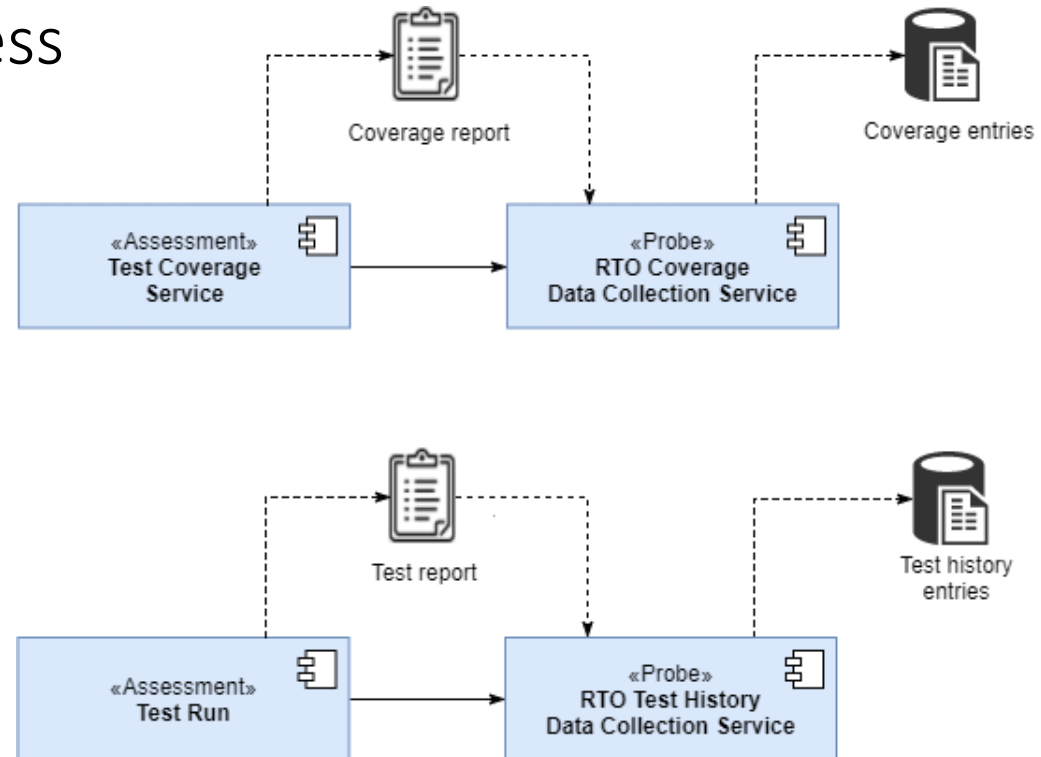
## “Context awareness”

Stored data can be identified by the producing delivery process and artifact(s)



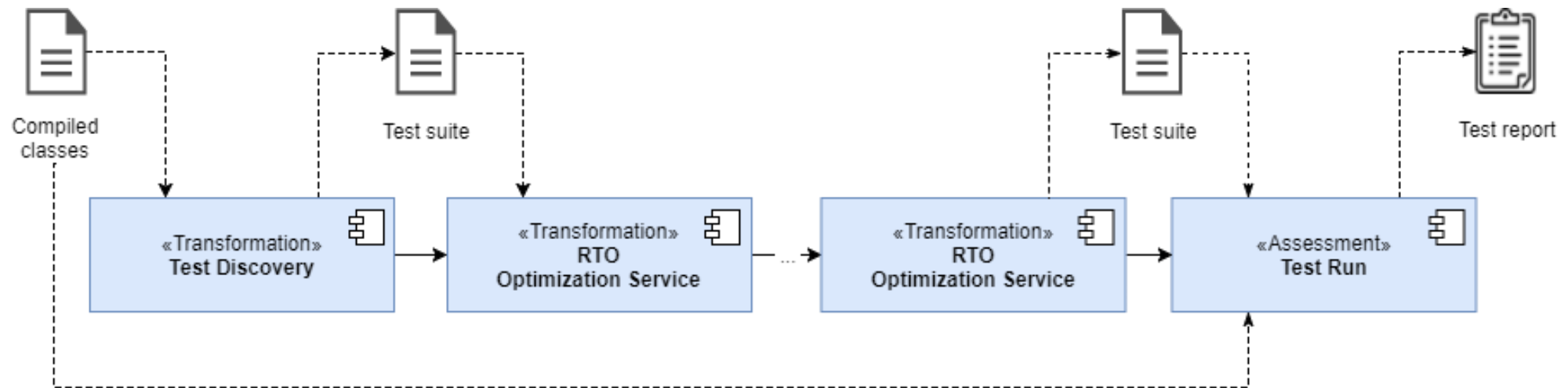
## Probe: Data Collection Examples

- Flexibility of modeling data collection
  - Can be placed anywhere during the delivery process
  - Independent from RTO process



## RTO Service Workflow

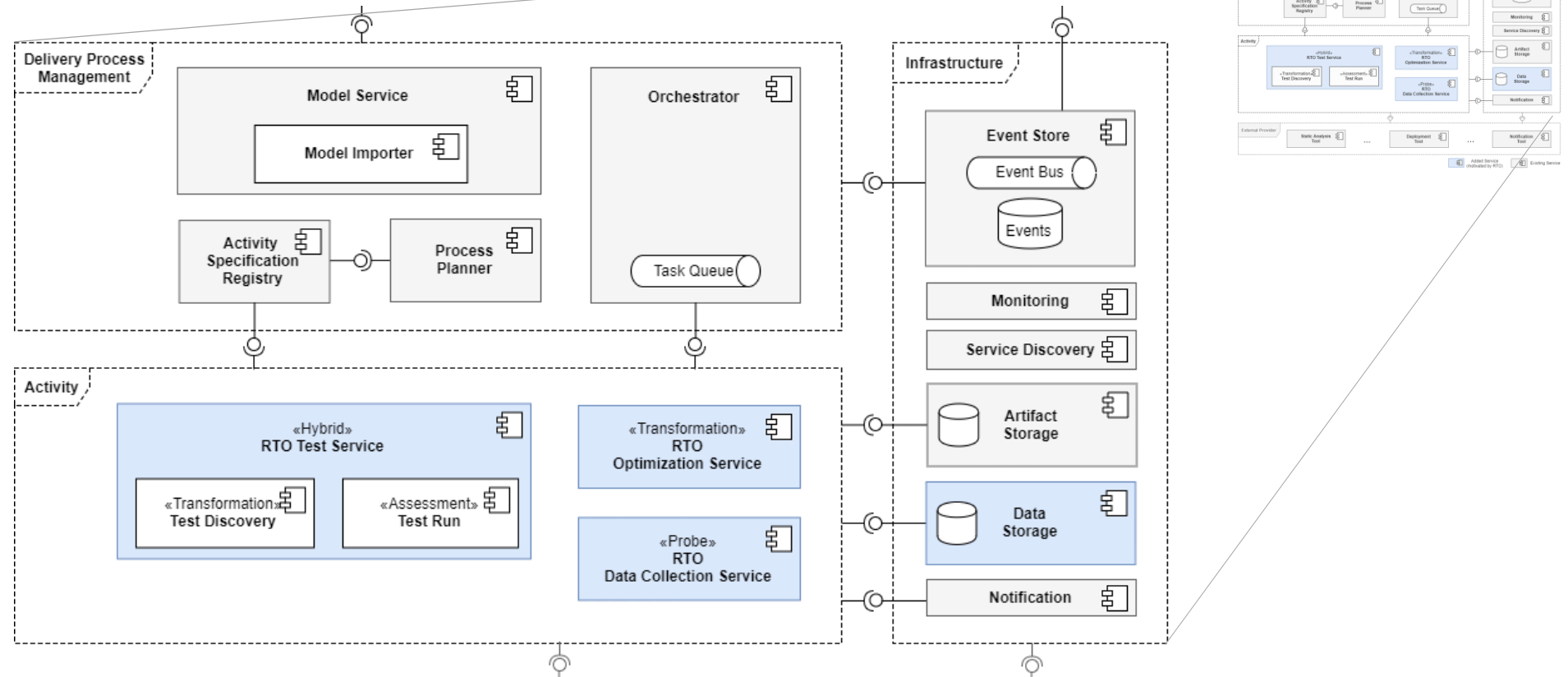
- RTO Pipeline is ultimately moved to the delivery process





# RTO in Jarvis Architecture

- “Local impact”
- Changes limited to activity and infrastructure layer



# RTO Pipeline in Delivery Model

```
- name: rto-dist
  service: rto-test-service
  command: discover
  dependsOn:
    - alias: compile
      ref: p://this/transformations/compile/classes
  parameters:
    classes: "@compile"

- name: rto-optimize-testtime
  service: rto-optimize-testexecutiontimeprioritization-service
  command: optimize
  dependsOn:
    - alias: testclasses
      ref: p://this/transformations/rto-dist/testClasses
    - alias: model
      ref: p://this/model/id
  parameters:
    testRef: "@testclasses"
    modelId: "@model"

- name: rto-optimize-failedfirst
  service: rto-optimize-failedfirstprioritization-service
  command: optimize
  dependsOn:
    - alias: testclasses
      ref: p://this/transformations/rto-optimize-testtime/testRef
    - alias: model
      ref: p://this/model/id
  parameters:
    testRef: "@testclasses"
    modelId: "@model"
```

```
assessments:
  - name: rto-run
    service: rto-test-service
    command: execute
    dependsOn:
      - alias: compile
        ref: p://this/transformations/compile/classes
      - alias: testclasses
        ref: p://this/transformations/rto-optimize-failedfirst/testRef
    parameters:
      classes: "@compile"
      testOrder: "@testclasses"


probes:
  - name: rto-probe
    service: rto-test-history-probe-service
    command: probe
    dependsOn:
      - alias: testreport
        ref: p://this/assessments/rto-run/testReport
      - alias: model
        ref: p://this/model/id
    parameters:
      testReport: "@testreport"
      modelId: "@model"
```



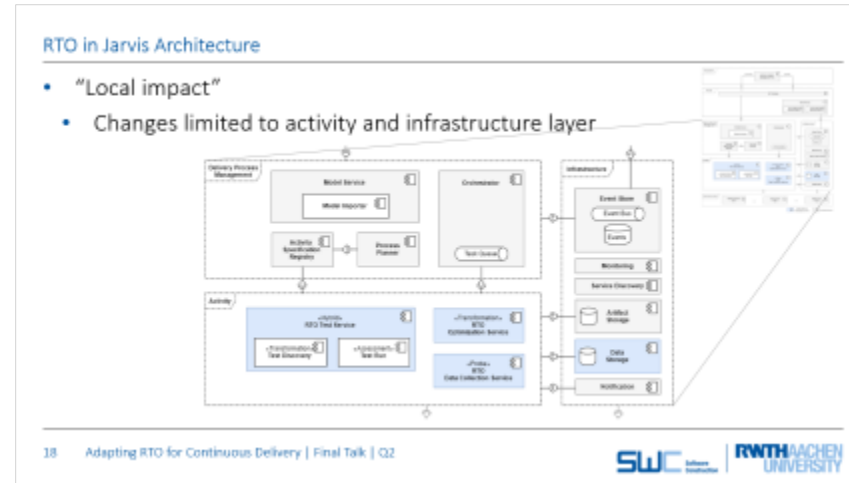
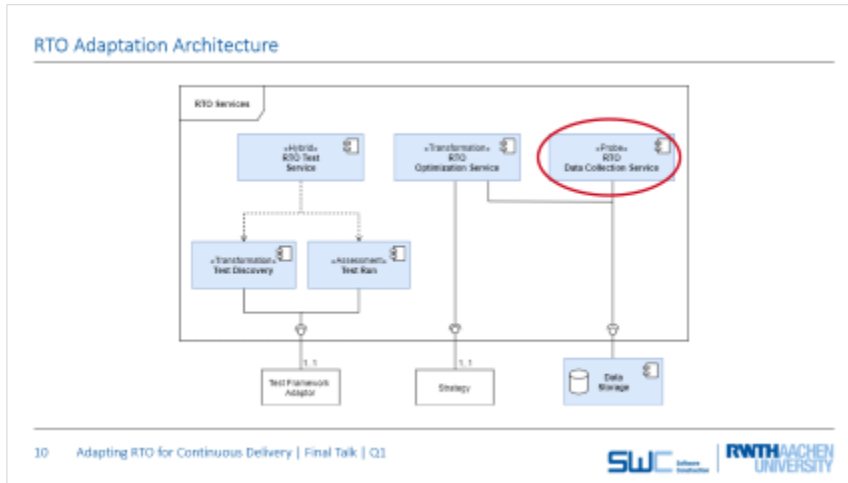
## Properties of RTO Adaptation

---

1. Independent data collection
2. Easy integration of new RTO techniques
3. Distributed RTO process – unclear boundary
  - Forced by the framework to decouple
  - But the services themselves are still coupled as a process
  - Internal complexity -> External modeling complexity

➤ Easy adoption of RTO in the delivery process -> just add to the model! 

# Summary



- ### Properties of RTO Adaptation
1. Independent data collection
  2. Easy integration of new RTO techniques
  3. Distributed RTO process – unclear boundary
    - Forced by the framework to decouple
    - But the services themselves are still coupled as a process
    - Internal complexity -> External modeling complexity
- Easy adoption of RTO in the delivery process -> just add to the model! 😊
- 21 Adapting RTO for Continuous Delivery | Final Talk | Q3
- SWC RWTH AACHEN UNIVERSITY