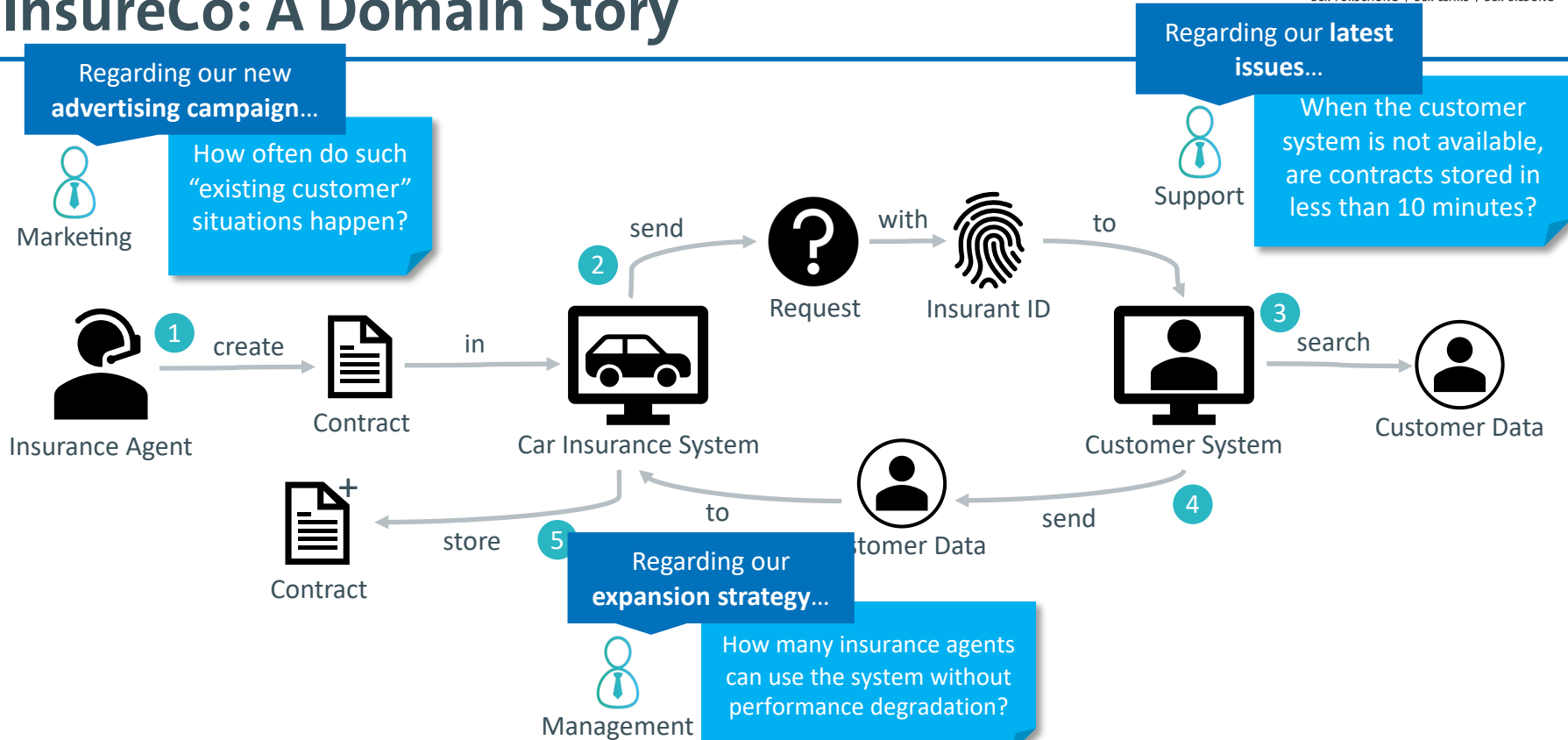


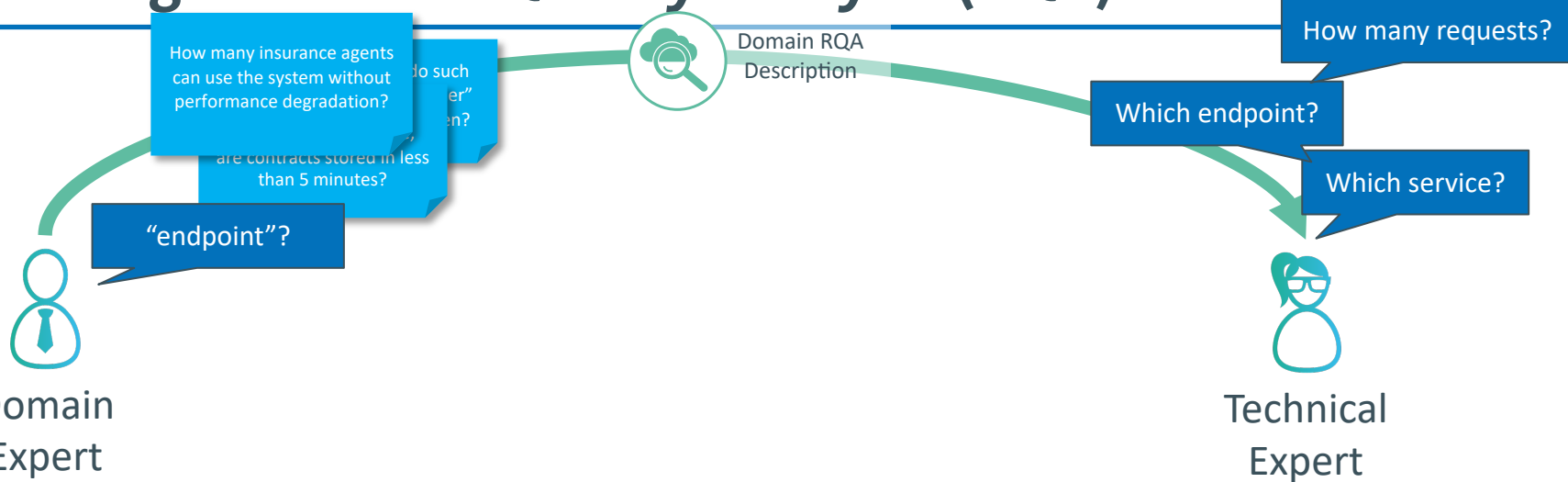
Sebastian Frank, Julian Brott, Dominik Kesim, Heiko Holz, Matthias Eschhold, André van Hoorn

dqualizer: Domain-Centric Runtime Quality Analysis of Business-Critical Application Systems

InsureCo: A Domain Story



Performing a Runtime Quality Analysis (RQA)



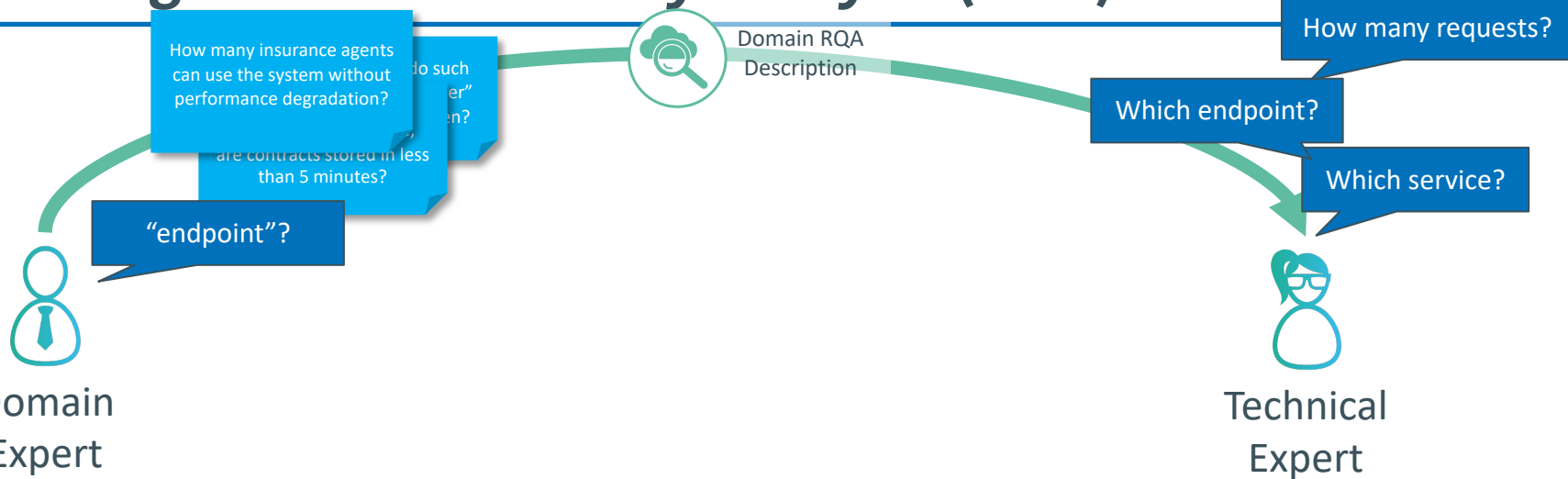
Business Knowledge



Technical Knowledge



Performing a Runtime Quality Analysis (RQA)



Business Knowledge



Domain Stories

Tactical Design

Domain Driven Design (DDD) [1]

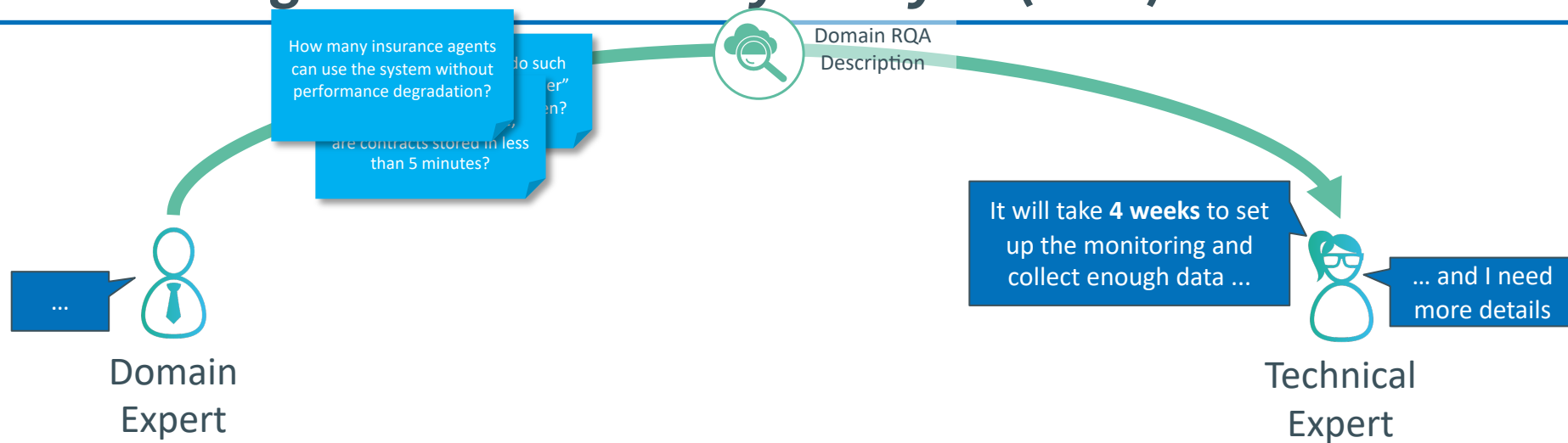
Ubiquitous Language

Strategic Design

Technical Knowledge



Performing a Runtime Quality Analysis (RQA)



Business Knowledge



Domain Stories

Tactical Design

Domain Driven Design (DDD) [1]

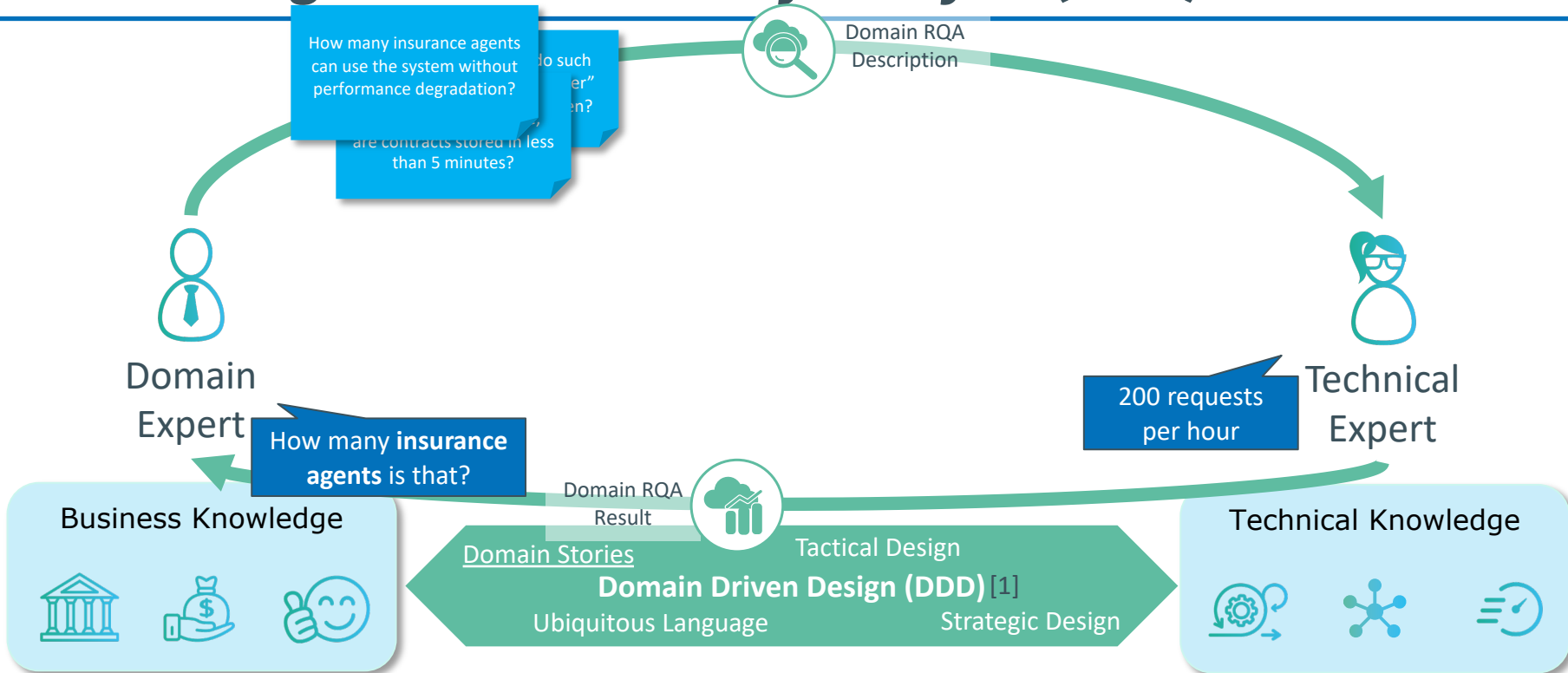
Ubiquitous Language

Strategic Design

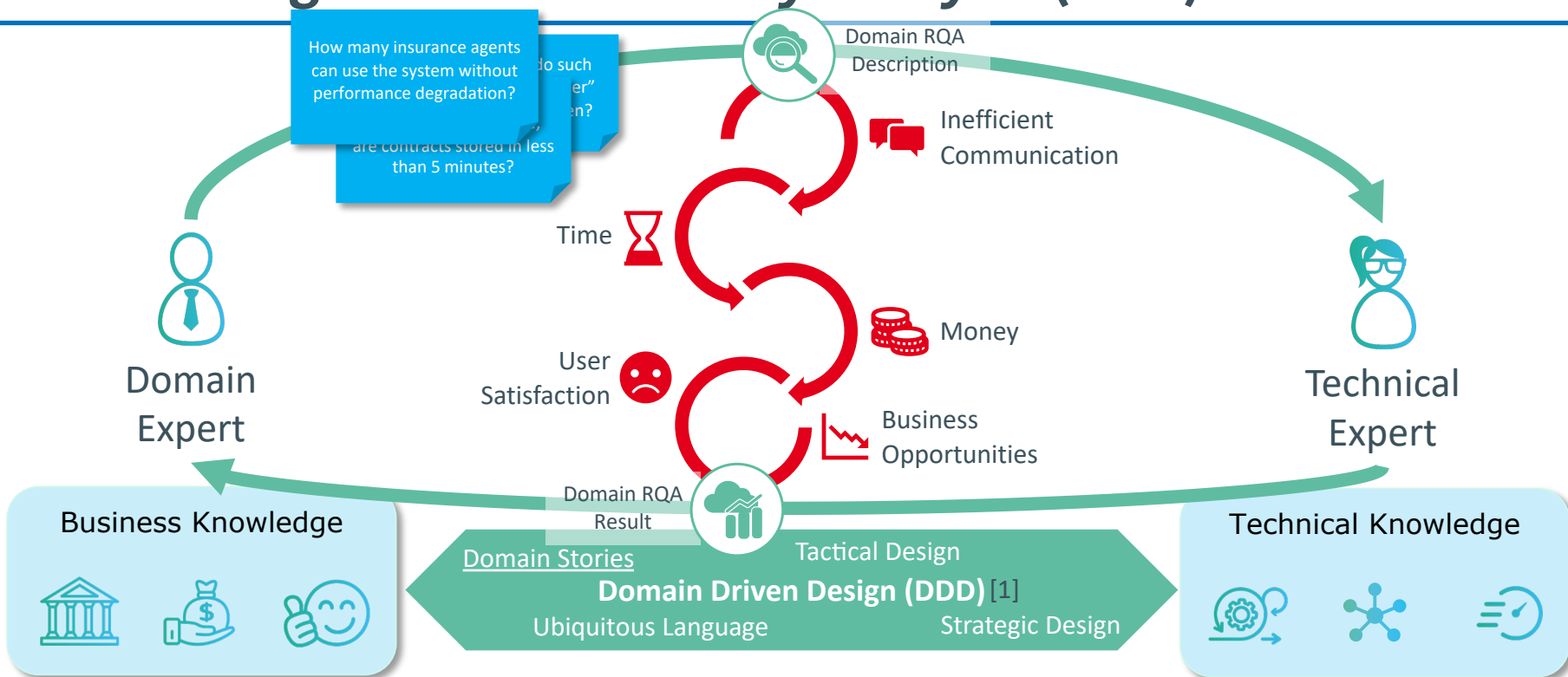
Technical Knowledge



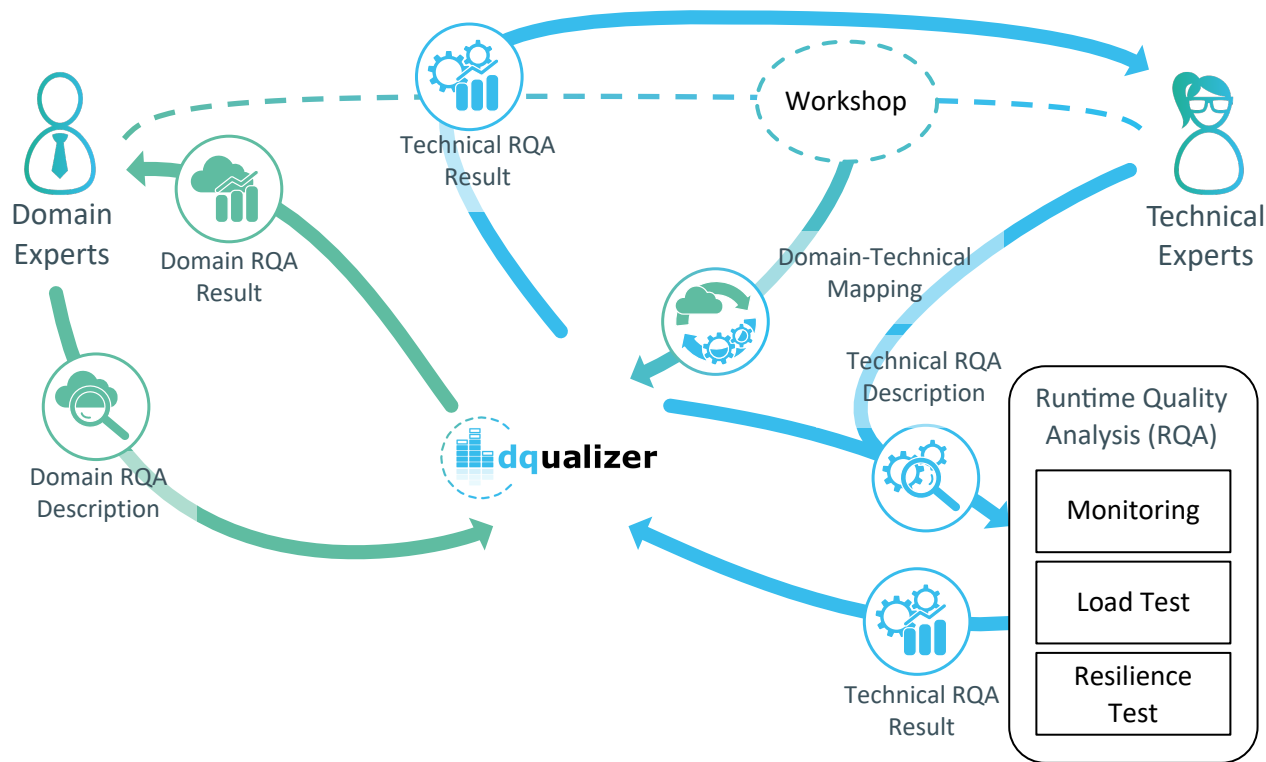
Performing a Runtime Quality Analysis (RQA)



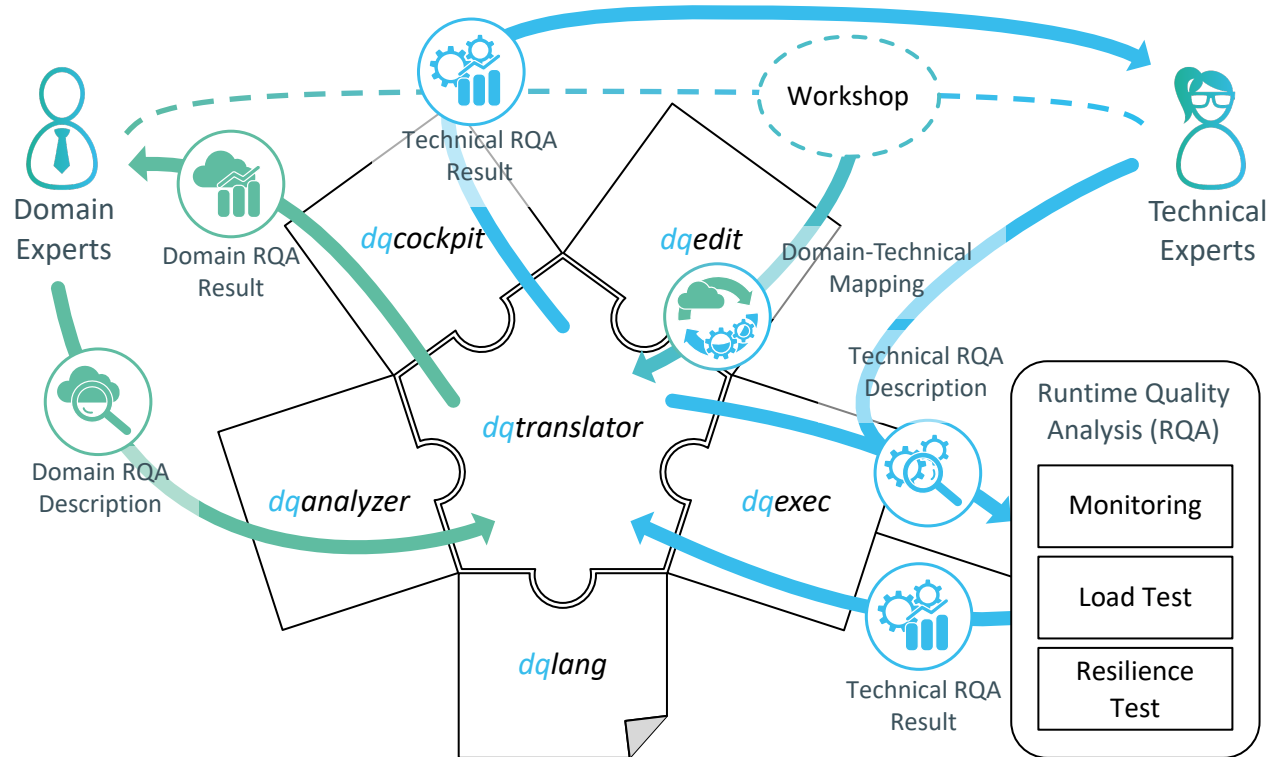
Performing a Runtime Quality Analysis (RQA)



dqualizer: Vision of Domain-Centric RQA

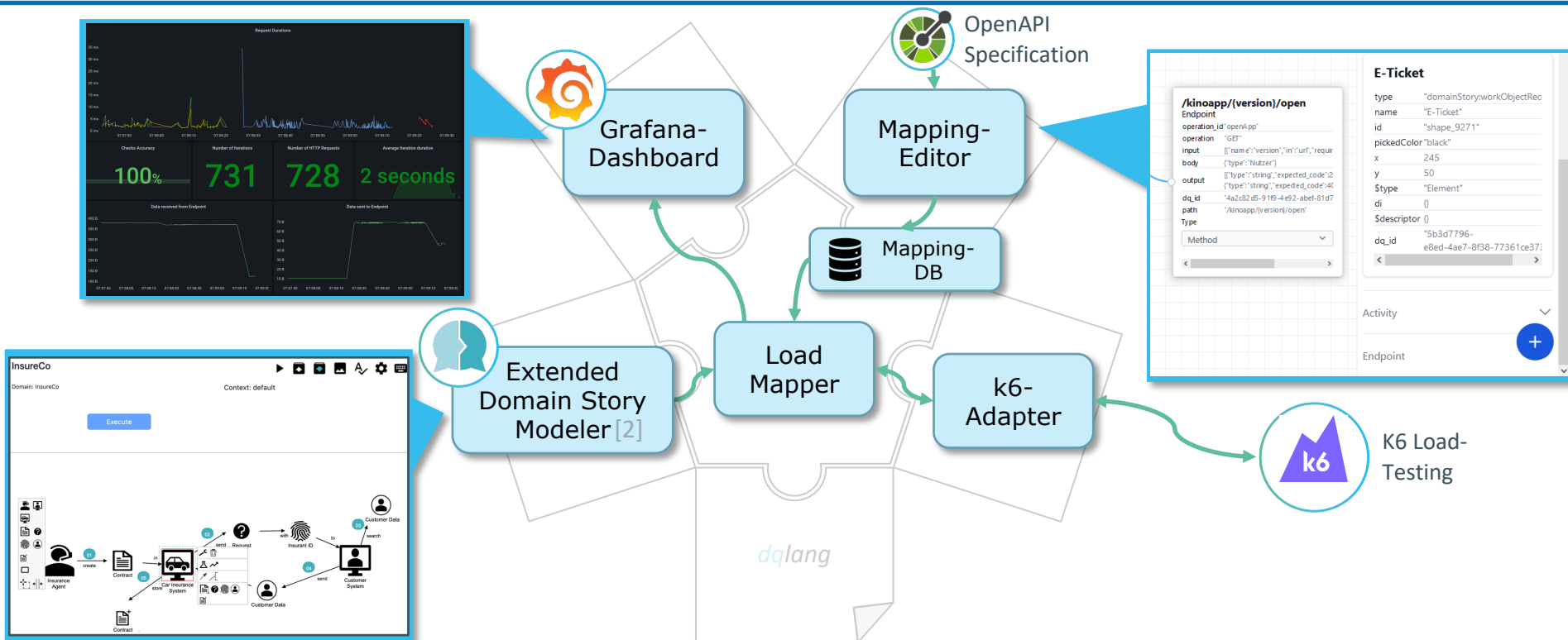


dqualizer: Vision of Domain-Centric RQA



[2] <https://egon.io/>
 Universität Hamburg
 DER FORSCHUNG | DER LEHRE | DER BILDUNG

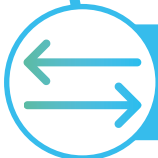
dqualizer: Current State



dqanalyzer: Prototype



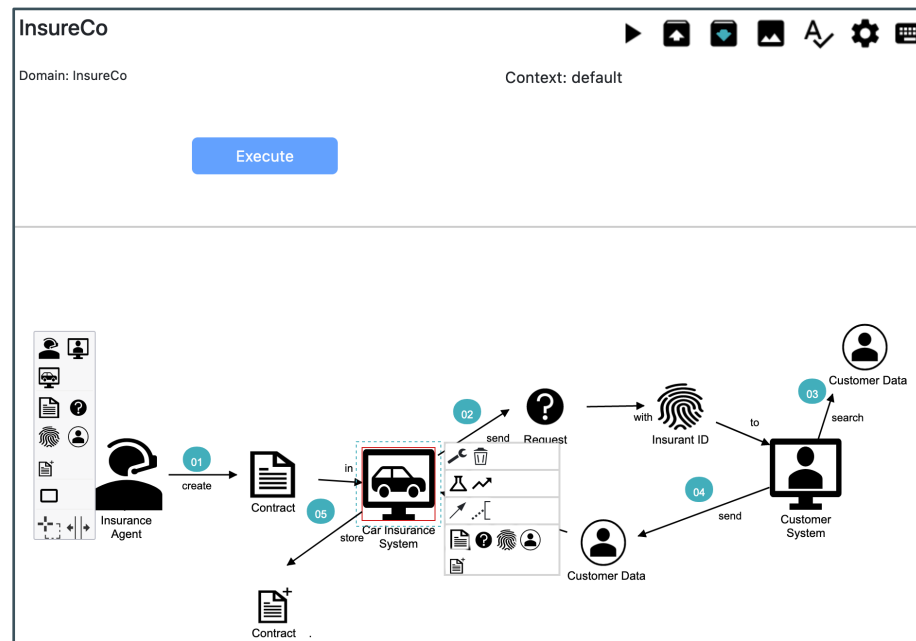
How can we enable domain experts to **model RQAs** using Domain Storytelling?



How can we **transform the annotated domain story** to a resilience and load test (conceptually)?



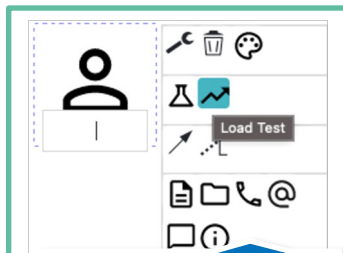
How can we **report the RQA results** on the domain-level to the domain experts?



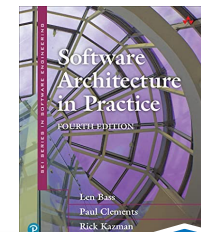
[3] *Apdex Users Group 2023. Apdex Performance Index. <https://www.apdex.org>.*

[4] *Len Bass, Paul Clements, and Rick Kazman. 2021. Software Architecture in Practice (4 ed.). Addison-Wesley Longman Publishing Co., Inc., USA.*

dqanalyzer: Concept



Annotations



Scenario Format

Satisfied $T < 1.2$ ms

Tolerated

$T > 4.8$ ms Frustrated

Apdex Levels

[3]

Chaos Toolkit

Alt. Terminology

target-service

artifact

fault repetition

occurrence

...

...



Chaos Toolkit

Alt. Terminology

[5] Dušan Okanović, André van Hoorn, Christoph Zorn, Fabian Beck, Vincenzo Ferme, and Jürgen Walter. 2019. Concern-driven reporting of software performance analysis results. In *Companion of the 2019 ACM/SPEC International Conference on Performance Engineering*. ACM, 1–4.

dqanalyzer: Result Presentation

based on [5]

RQA Summary

Resilience Results

We executed the resilience test with the stimulus **Failed request** using Chaos Toolkit, in the environment **TESTING**. You stated that the test should be executed **during regular office hours, i.e., between 08:00 am and 16:00 pm**. You also added **existing load tests to simulate real user behavior**. The stimulus was repeated **more than once**. The test results should have an confidence of **100%**. As a hypothesis you stated the **error rate** to be **low**.

Congrats! The artifact's (Employee Application) error rate is below the threshold you specified! The actual error rate was 4.2%! :-)

Close

RQA Result

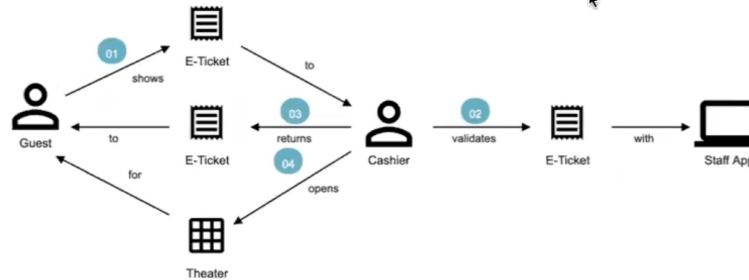


dqanalyzer: Demo

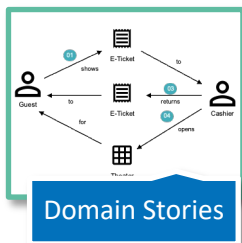
StoryCinemaEntranceControl

Domain: Metropolis

Context: default



Evaluation: Qualitative User Study

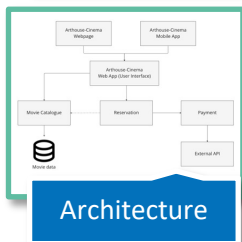


Modified Arthouse Cinema
 - Ticket Sale
 - Entrance Control

[6]



2x Insurance Domain
 2x Tax Domain

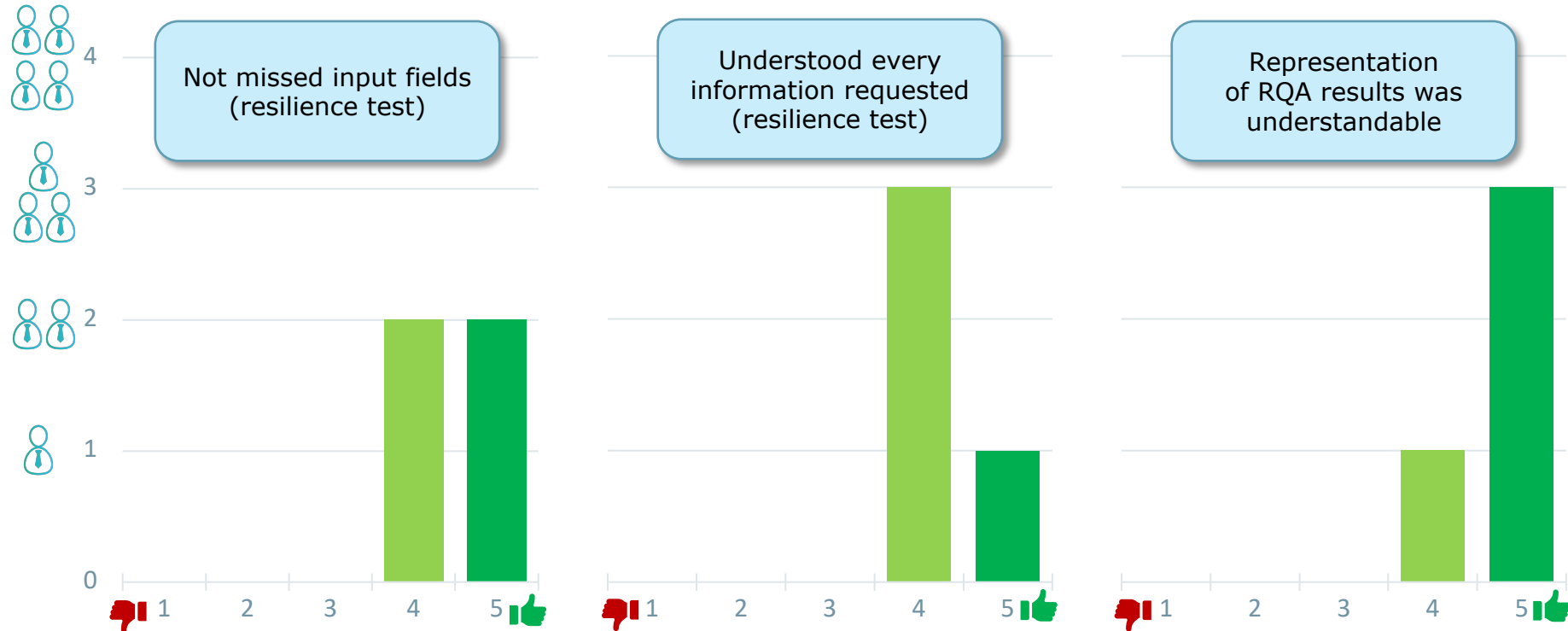


Component Diagram

- Resilience Test
- Load Test
- Resilience Test + Load
- Open Task



Evaluation: Selected Results



Evaluation: Lessons Learned

Domain experts can model their business questions easily and understand the representation of the RQA results, but...



Information Fields

Accompany Apdex ratings with **actual reference values**

Difficult to **understand** some fields (load peak design, ...)

Result Presentation

More **facts** regarding test runs

More **compact** representation

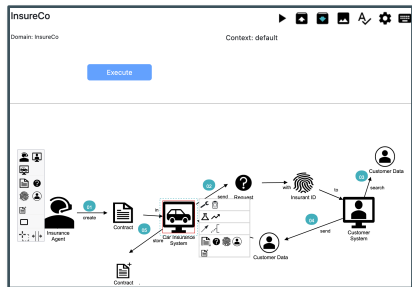
Summary suggests **interpretation**

Limitations

Putting RQA on **single element** of domain story

Limitations of Domain Story Modeler

Conclusion



Resilience Test

Artifact: Employee Application

Stimulus (*): Failed request

Confidence of Results (*): 100%

Response Measure (*):

Recovery time: Satisfied | Tolerated | Frustrated

Response time: Satisfied | Tolerated | Frustrated

Error rates: None | Low | Medium | High

Show loadtests

Save Close

Resilience Results

We executed the resilience test with the stimulus **Failed request** using Chaos Toolkit, in the environment **TESTING**. You stated that the test should be executed **during regular office hours, i.e., between 08:00 am and 16:00 pm**. You also added **existing load tests to simulate real user behavior**. The stimulus was repeated **more than once**. The test results should have a confidence of **100%**. As a hypothesis you stated the **error rate to be low**.

Congrats! The artifact's (Employee Application) error rate is below the threshold you specified! The actual error rate was **4.2%** :-)

Close

