



# A Software Architecture Framework for Quality-Aware DevOps

*D. A. Tamburri, E. Di Nitto, M. Guerriero,  
P. Jamshidi, I. Spais*

DICE

Horizon 2020 Research & Innovation Action

Grant Agreement no. 644869

<http://www.dice-h2020.eu>



Funded by the Horizon 2020  
Framework Programme of the European Union







- The ISO/IEC/IEEE 42010 Conceptual Model of Architecture Description<sup>[1]</sup> defines the term **architecture framework** as:

*“a (set of) **common practice(s)** for creating, interpreting, analyzing and using **architecture descriptions** within a particular **domain of application** or stakeholder community”*













- Comparative evaluation of previous well-known/established arch. Frameworks:
  - MODAF
  - RM-ODP
  - DODAF
  - TOGAF
  - 4+1-Views
  - ...

Systematic mapping of previous architecture frameworks<sup>[2]</sup>



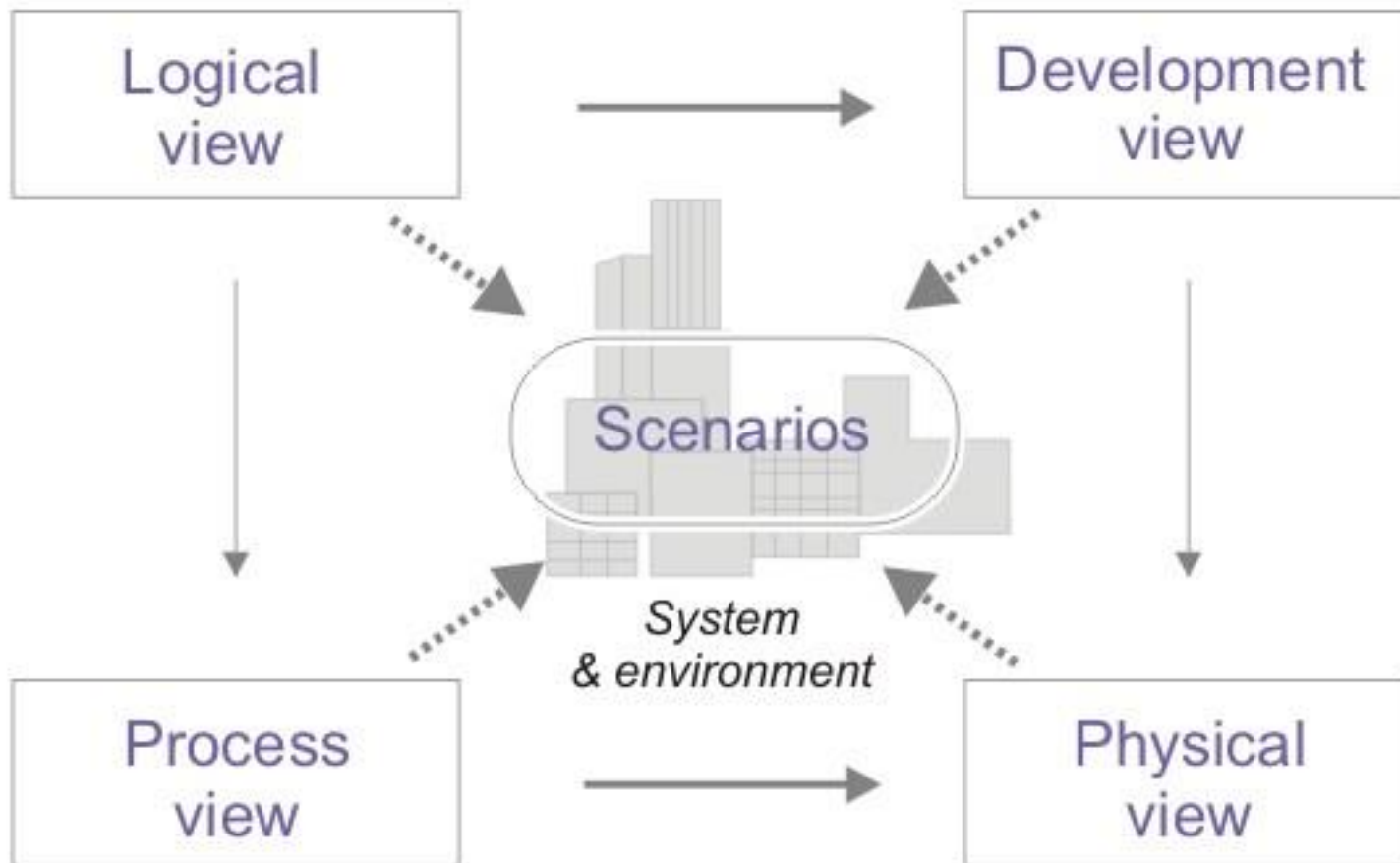
- Comparative evaluation of previous well-known/established arch. Frameworks:
  - MODAF
  - RM-ODP
  - DODAF
  - TOGAF
  - **4+1-Views**
  - ...

Systematic mapping of previous architecture frameworks<sup>[2]</sup>

# Quality-Aware DevOps Architecture Descriptions



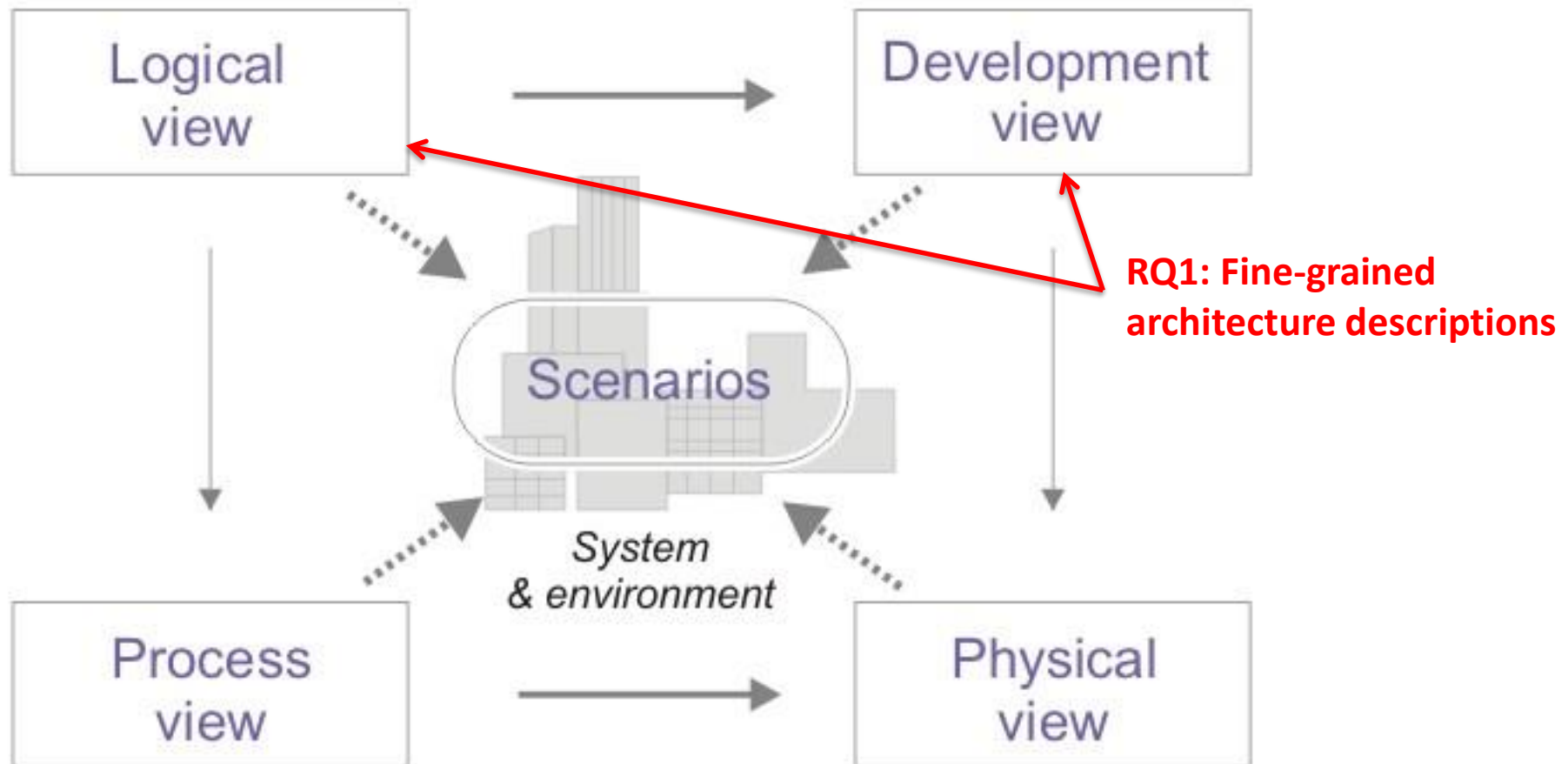
The 4+1 Views<sup>[3]</sup> model offers a basis consistent with the identified stakeholders and concerns for quality-aware DevOps...



# Quality-Aware DevOps Architecture Descriptions



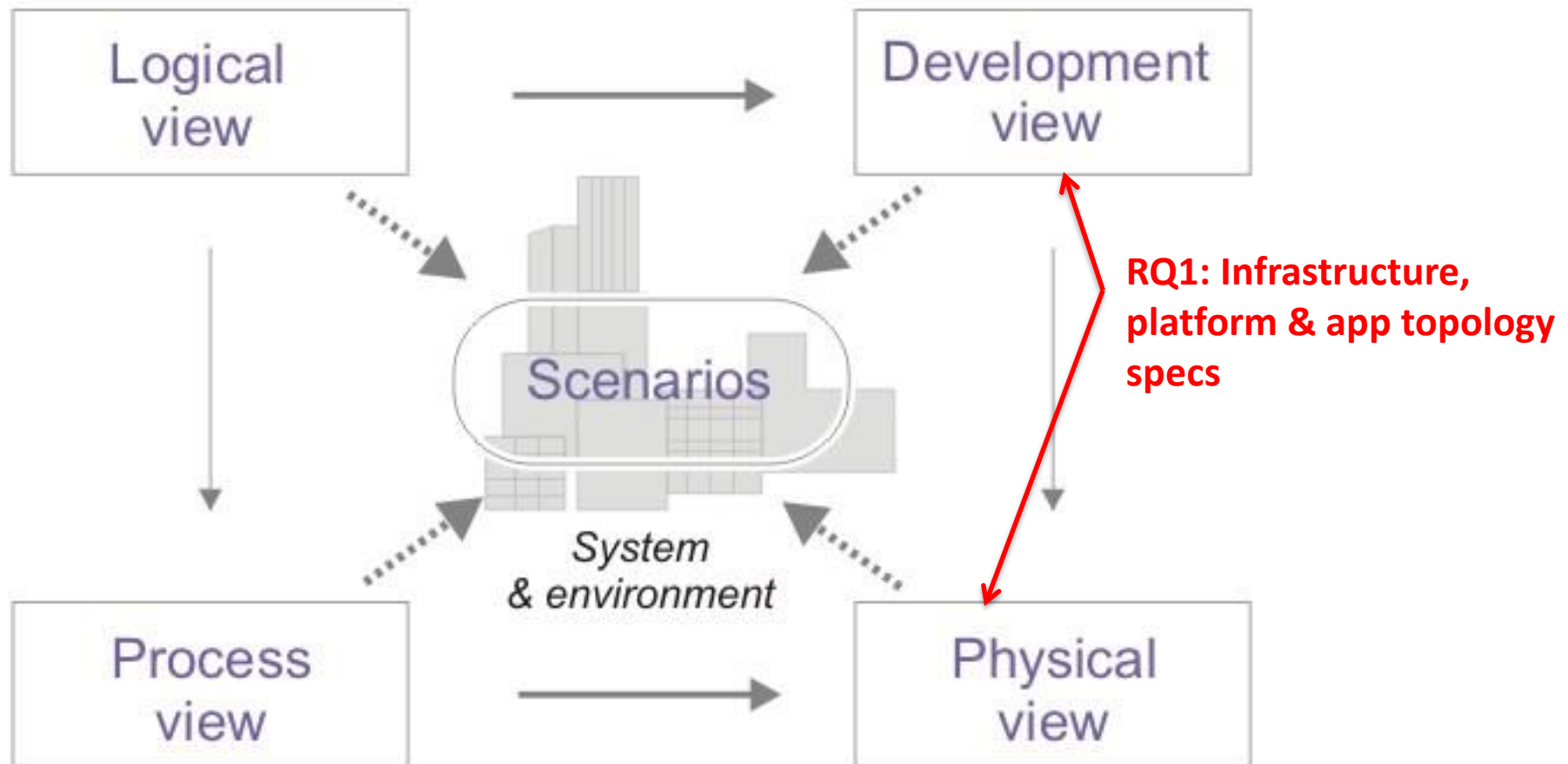
The 4+1 Views<sup>[3]</sup> model offers a basis consistent with the identified stakeholders and concerns for quality-aware DevOps...



# Quality-Aware DevOps Architecture Descriptions



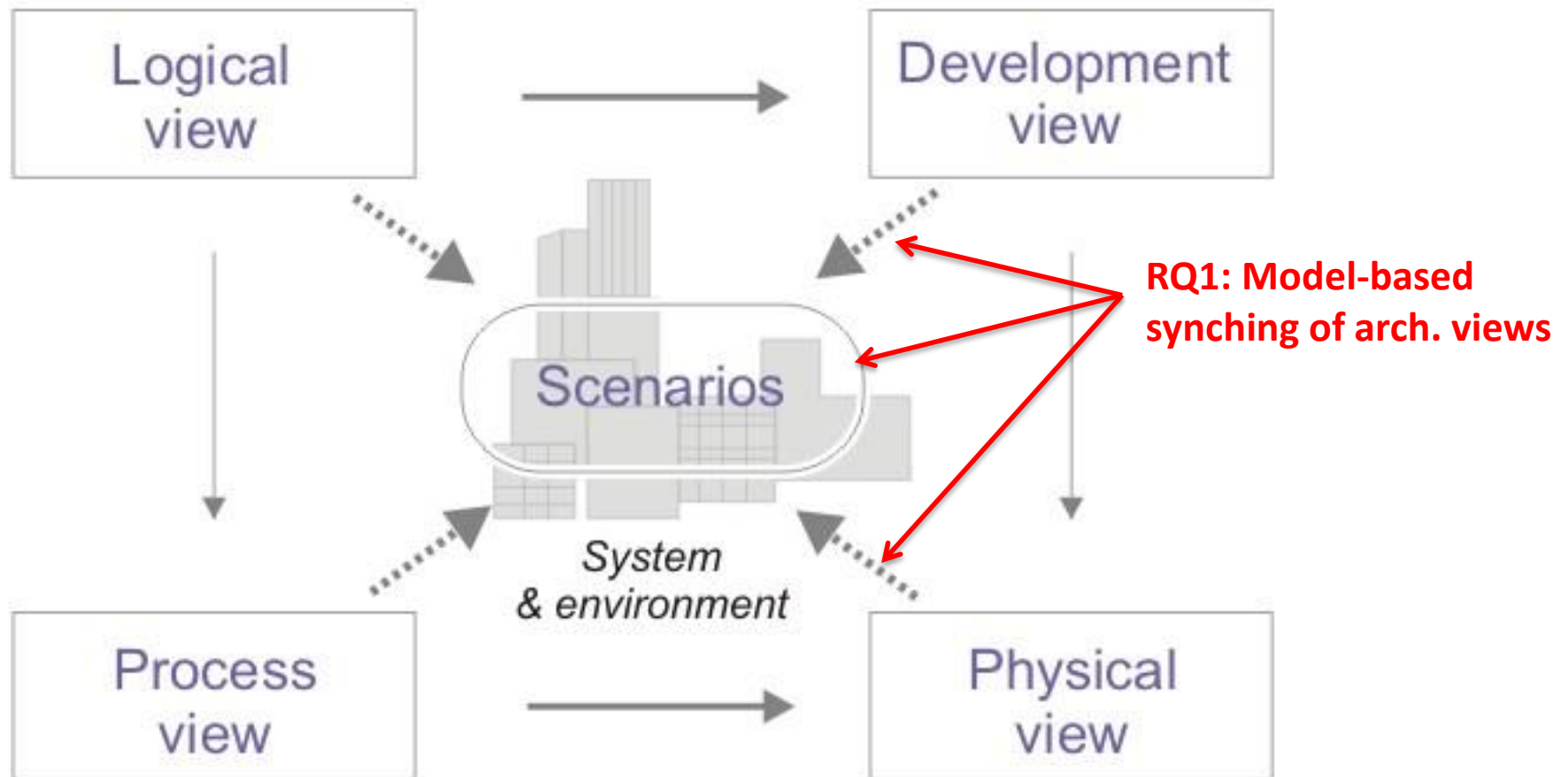
The 4+1 Views<sup>[3]</sup> model offers a basis consistent with the identified stakeholders and concerns for quality-aware DevOps...



# Quality-Aware DevOps Architecture Descriptions



The 4+1 Views<sup>[3]</sup> model offers a basis consistent with the identified stakeholders and concerns for quality-aware DevOps...





***But...***

<...moment of suspense...>





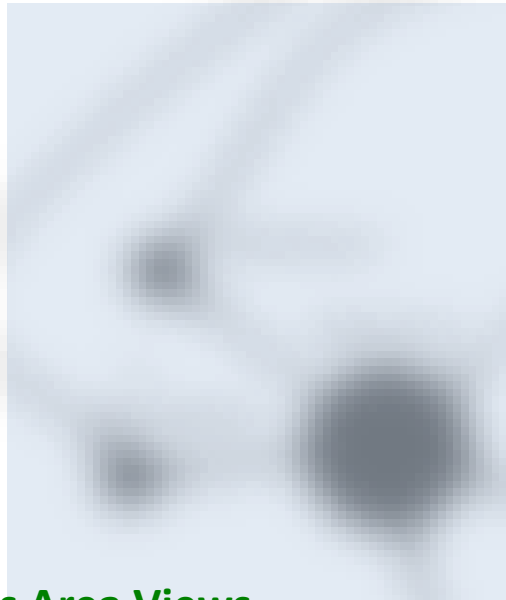
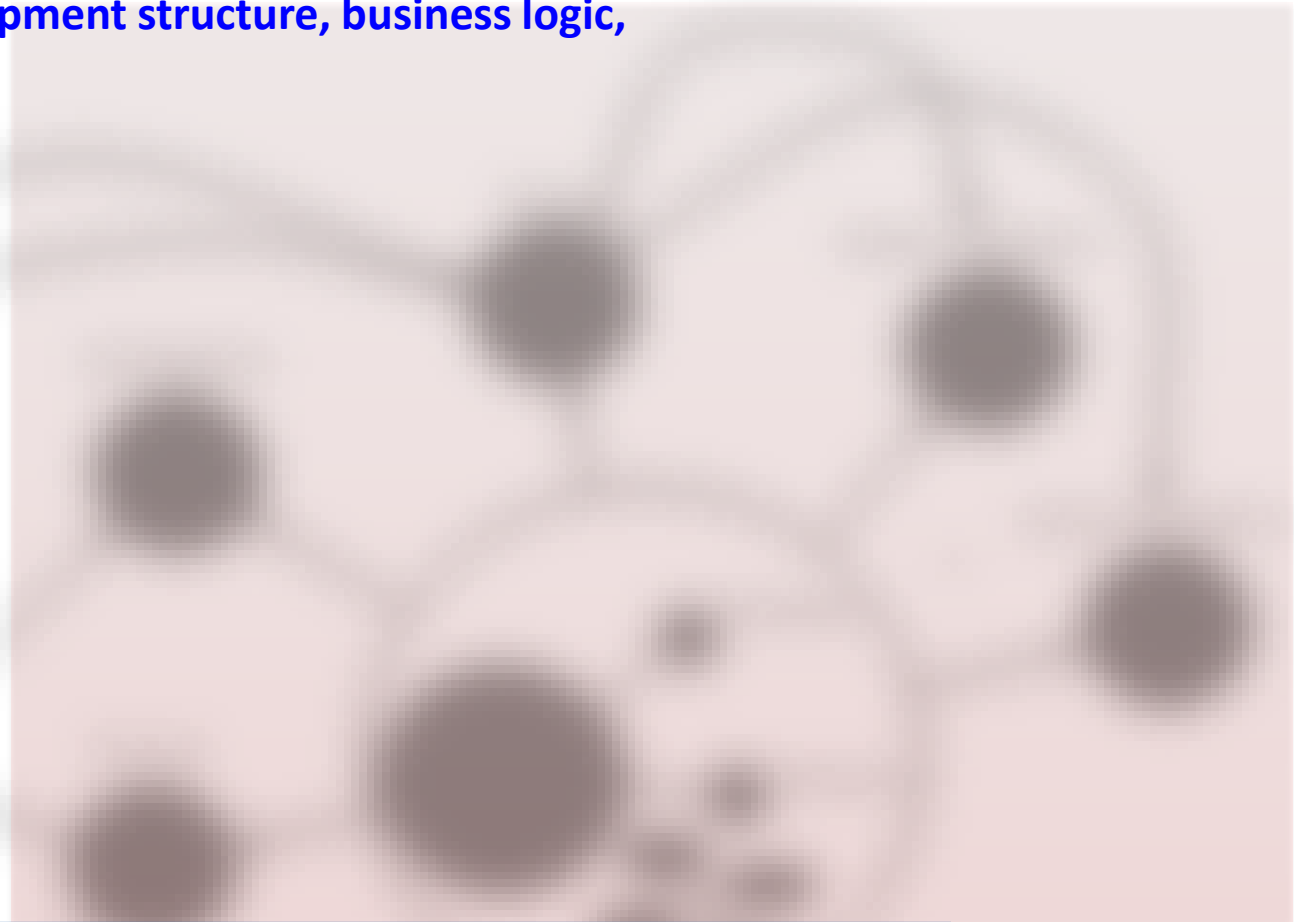




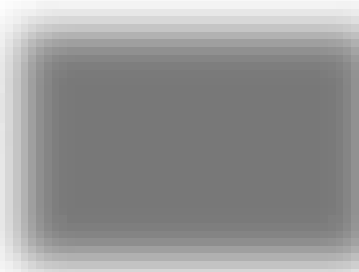
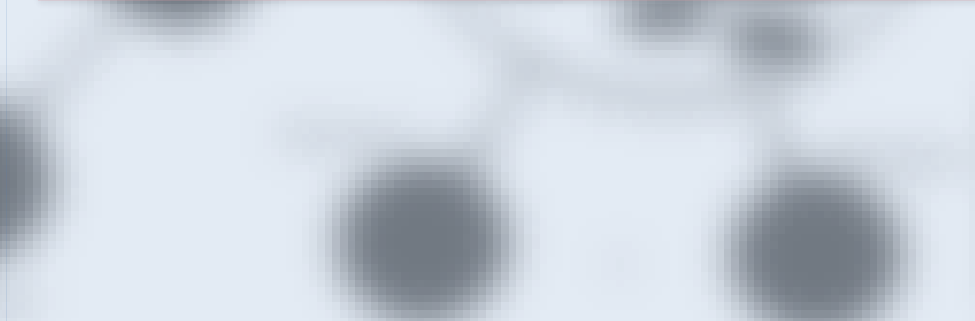
# Summing it all up: SQUID Views and Transformations



**Dev Area Views (Development structure, business logic, quality verification)**



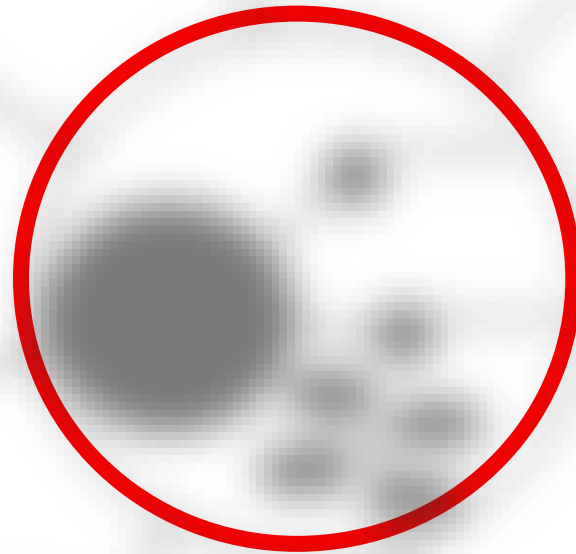
**Ops Area Views (Deployment, process, monitoring)**



# SQUID continuous architecting in action



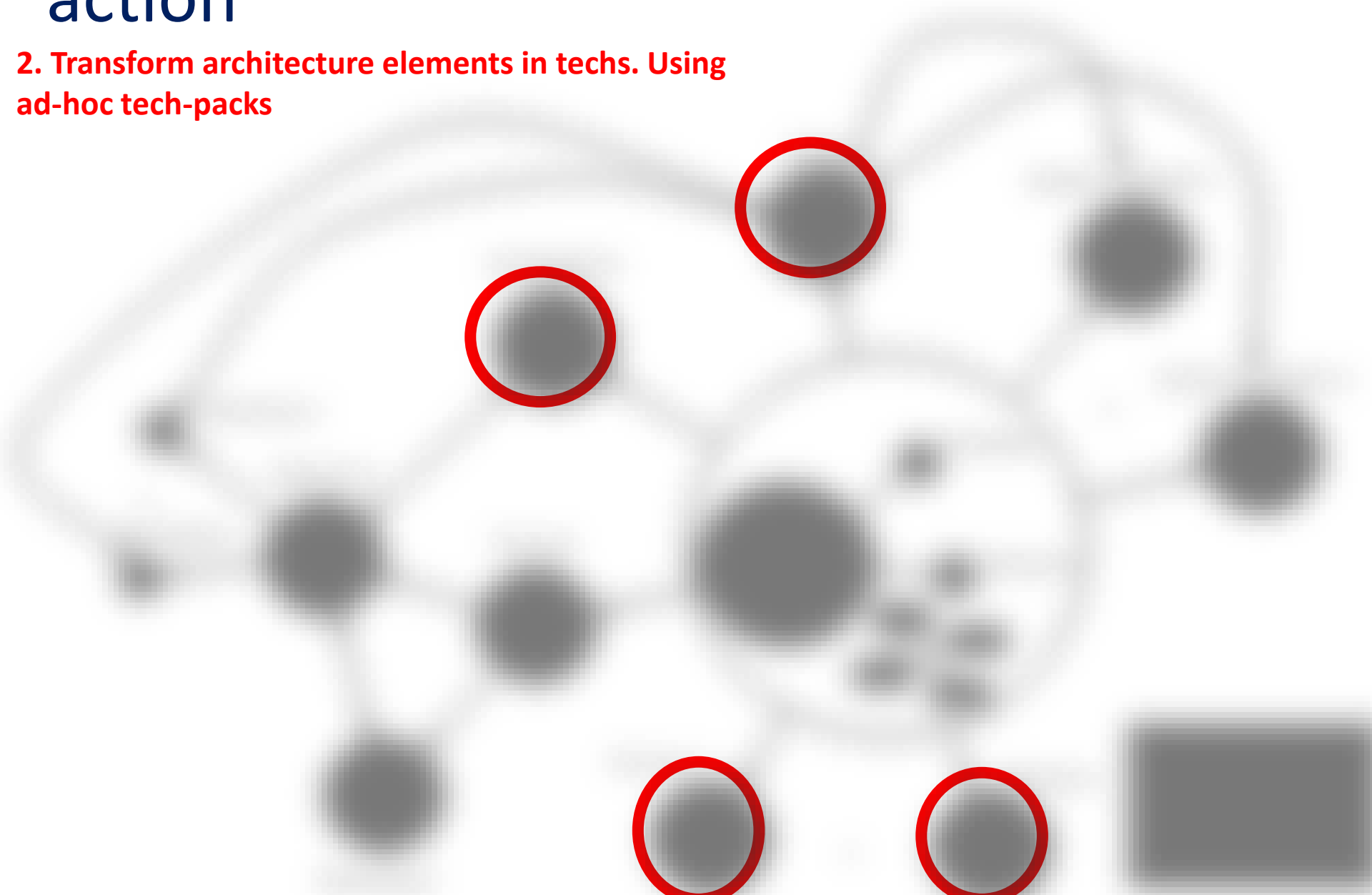
**1. Specify Architecture components and restrictions... UML profile!**



# SQUID continuous architecting in action



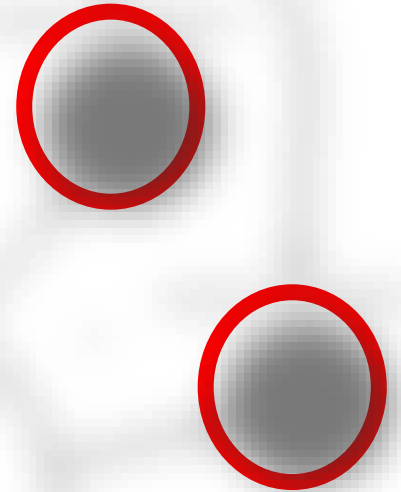
**2. Transform architecture elements in techs. Using ad-hoc tech-packs**



# SQUID continuous architecting in action



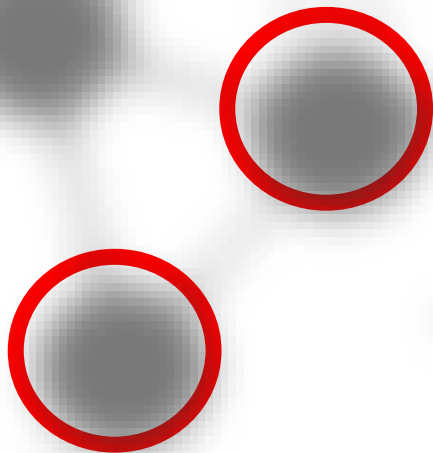
**3. Verify quality properties by means of ad-hoc annotations and tools**



# SQUID continuous architecting in action



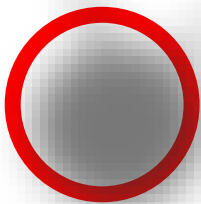
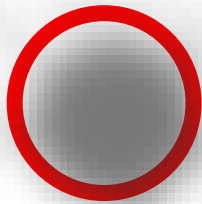
## 4. Deploy improved architecture



# SQUID continuous architecting in action

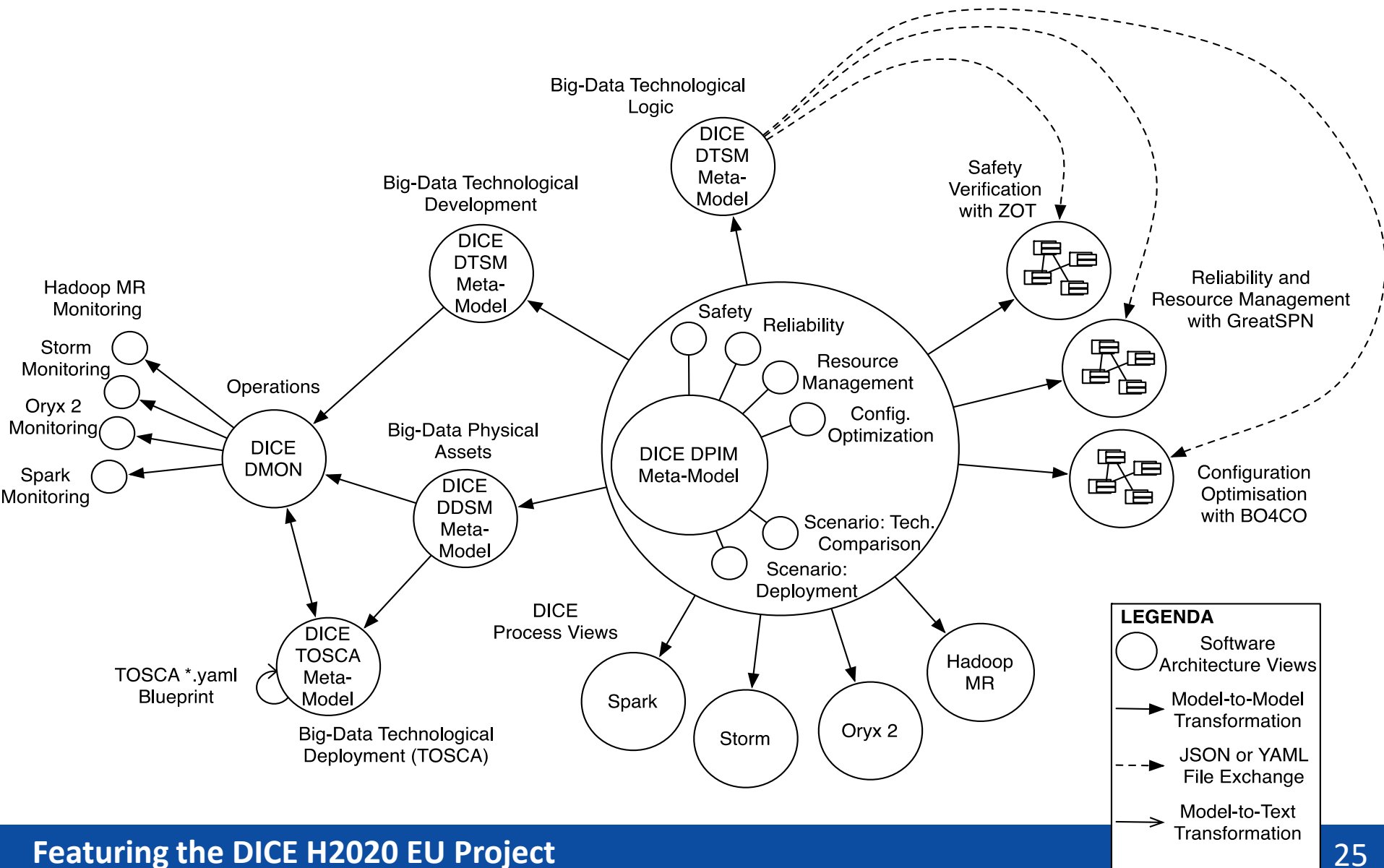


## 5. Monitor and continuously enhance/architect/(re-)deploy





# For example\* ...



# In conclusion... Take-home messages!



- MDE and DevOps are made for each other
  - SQUID offers a complete approach for Quality-Aware MDE-based continuous architecting (e.g., of DIAs)
  - Heavy use of M2M and M2T transformations
- Needs in DevOps rotate around multi-view and continuous-architecting
  - Speedy modeling, synch and (re-)deployment are critical



- [1] Group, I. A. W. (2000), 'IEEE Std 1471-2000, Recommended practice for architectural description of software-intensive systems' , Technical report, IEEE , IEEE , i--23 .
- [2]<http://www.iso-architecture.org/42010/afs/frameworks-table.html>
- [3] Kruchten, P. (1995), 'Architectural Blueprints: The "4+1" View Model of Software Architecture', *IEEE Software* **12** (6), 42-50.