



# DICE: Quality-Aware DevOps For Big Data Applications

Giuliano Casale  
*Imperial College London*

DICE Horizon 2020 Project  
Grant Agreement no. 644869  
<http://www.dice-h2020.eu>

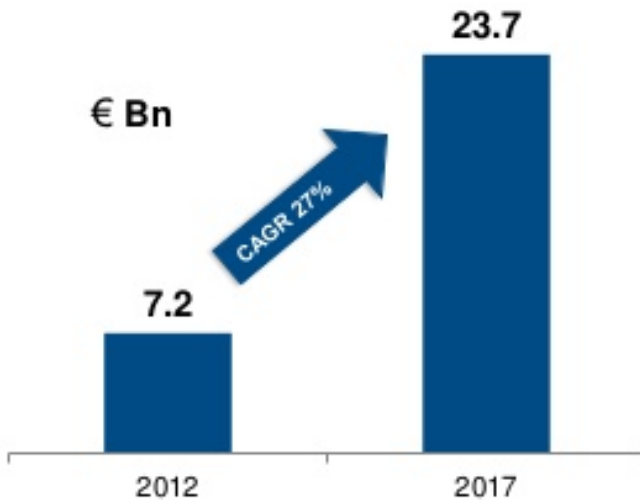


Funded by the Horizon 2020  
Framework Programme of the European Union

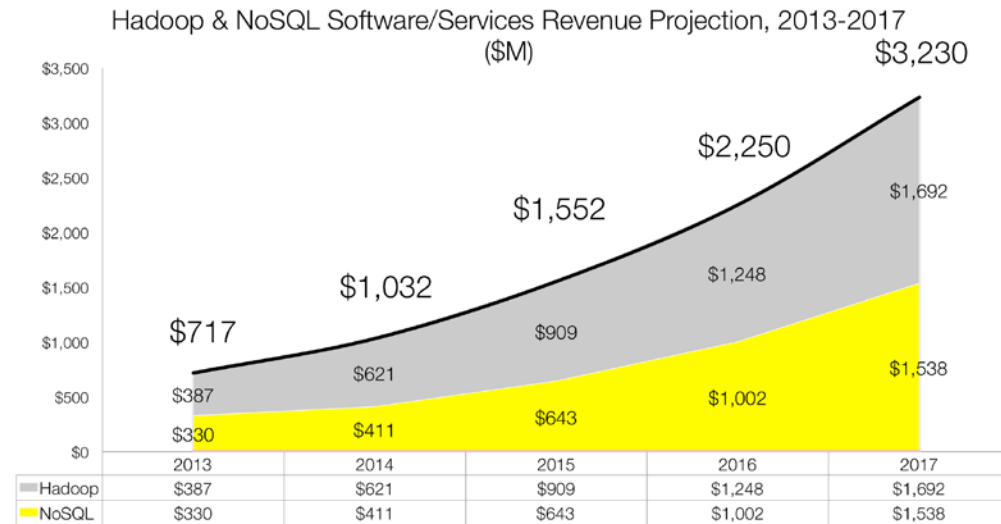
# The Rapid Growth of Big Data



- Software market rapidly shifting to Big data
  - 27% compound annual growth rate through 2017 (IDC)
  - Popular technologies such as Spark, Hadoop, and NoSQL boost Big Data adoption and revenues from new services



Source: IDC



Source: Wikibon

Business issue: 65% of Big data projects still fail (CapGemini)

# What problems EU SMEs face?



An example from our consortium



Traditional market:  
Legacy software systems

*Learning curves*



*Initial prototype*



*Risk of failure*



*Fast-paced market*



Customers with legacy data now ask for Big Data technologies



Growth in sight, but ...



# DICE: DevOps for DIAs



*Mission:* support SMEs in developing high-quality cloud-based data-intensive applications (DIAs)

- Horizon 2020 research project (4M€, 2015-18)
- 9 partners (Academia & SMEs), 7 EU countries

Imperial College  
London



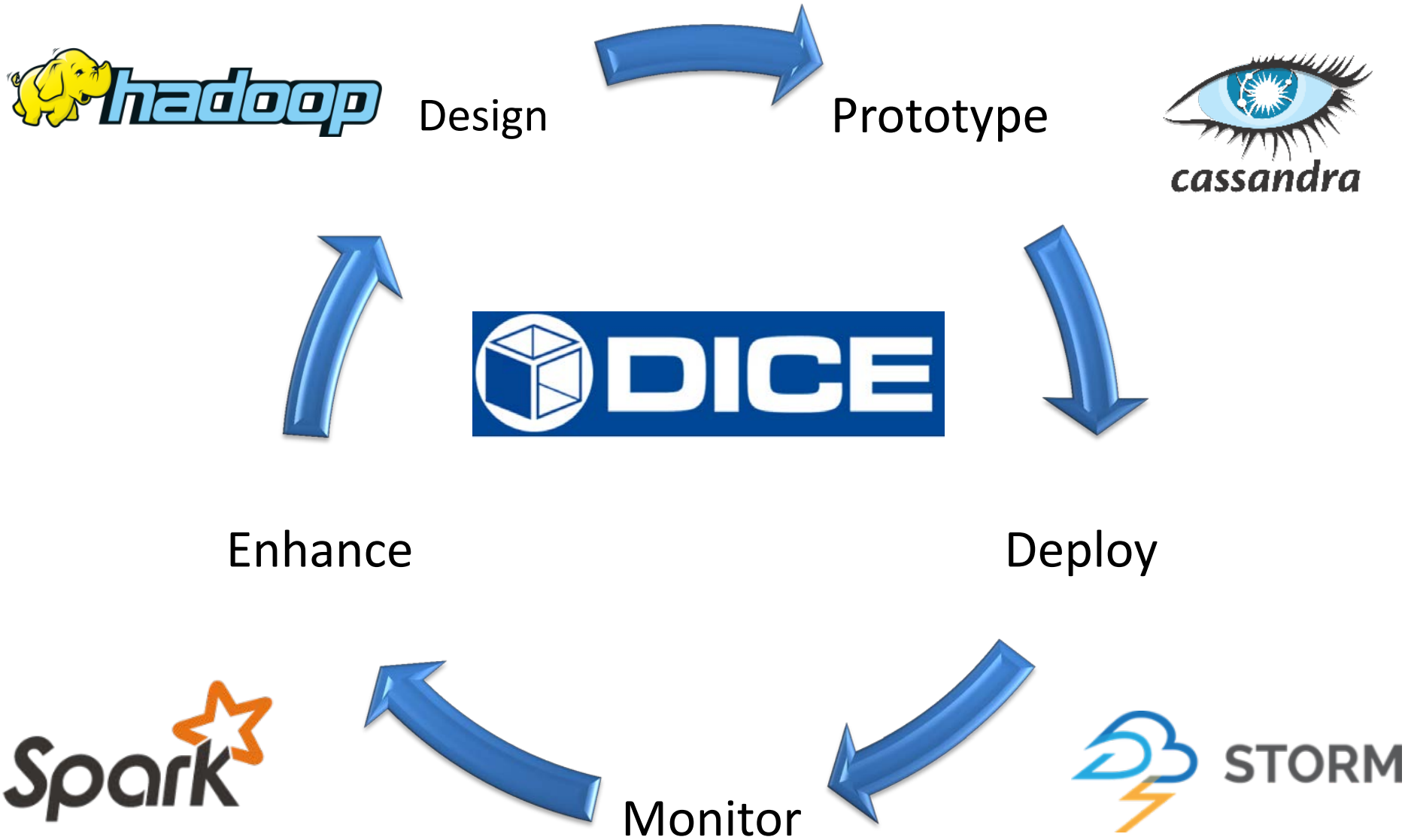
Universidad  
Zaragoza



POLITECNICO  
DI MILANO



# DICE: Quality-Aware DevOps for Big Data

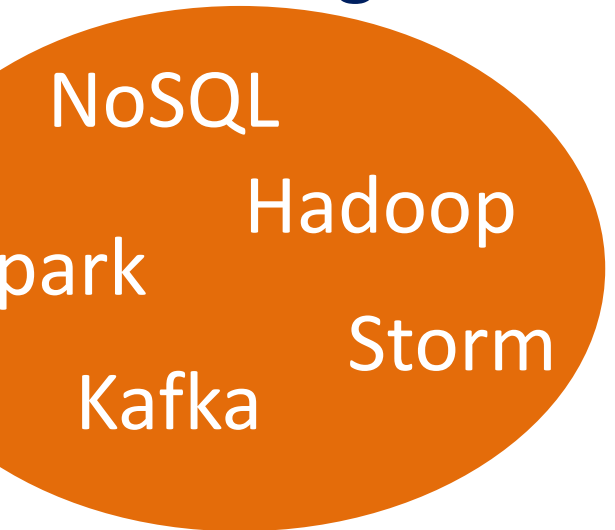


# How to support DIA development?

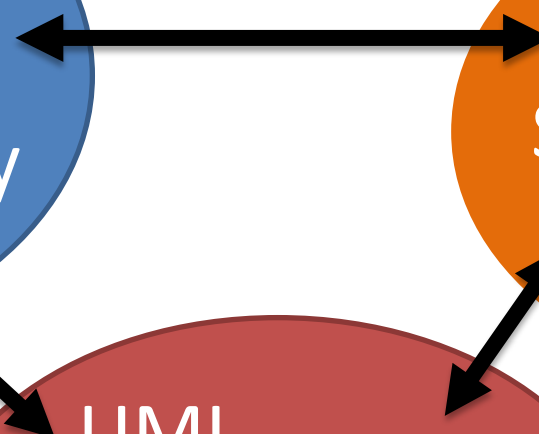
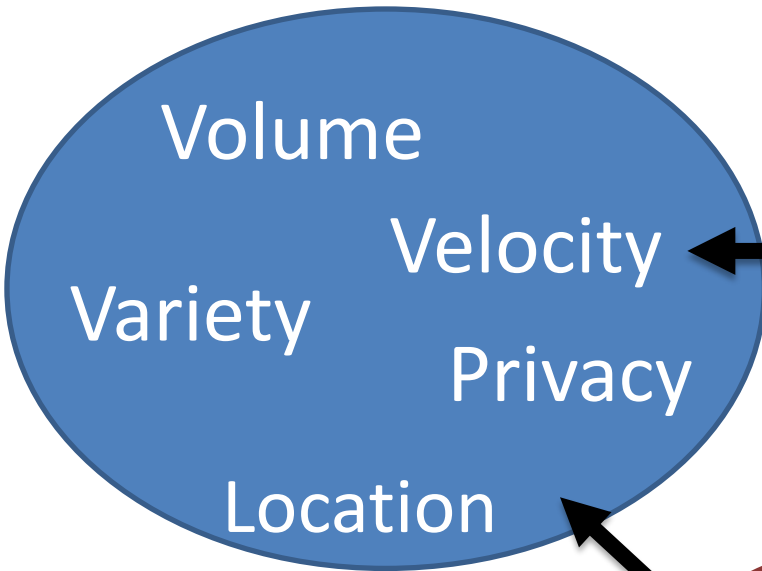
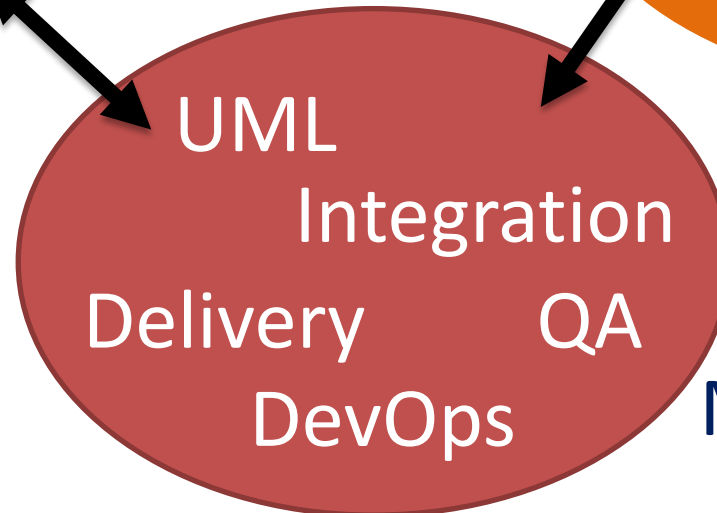


## Characterize Data Properties

## Big Data Technologies



## Development Methods & Tools



# What do we mean by Quality?



○ Reliability

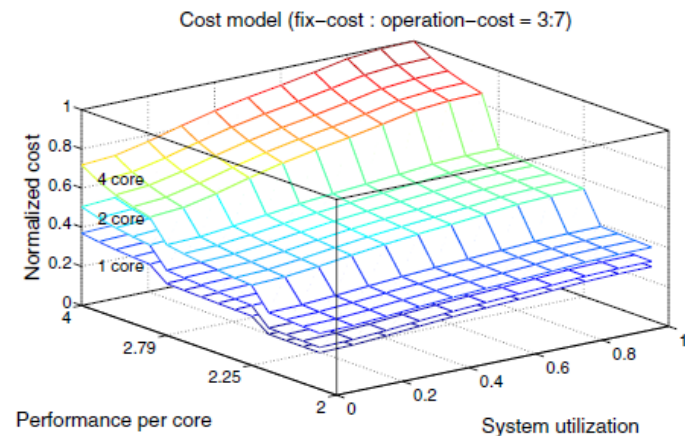
- Availability
- Fault-tolerance

○ Efficiency

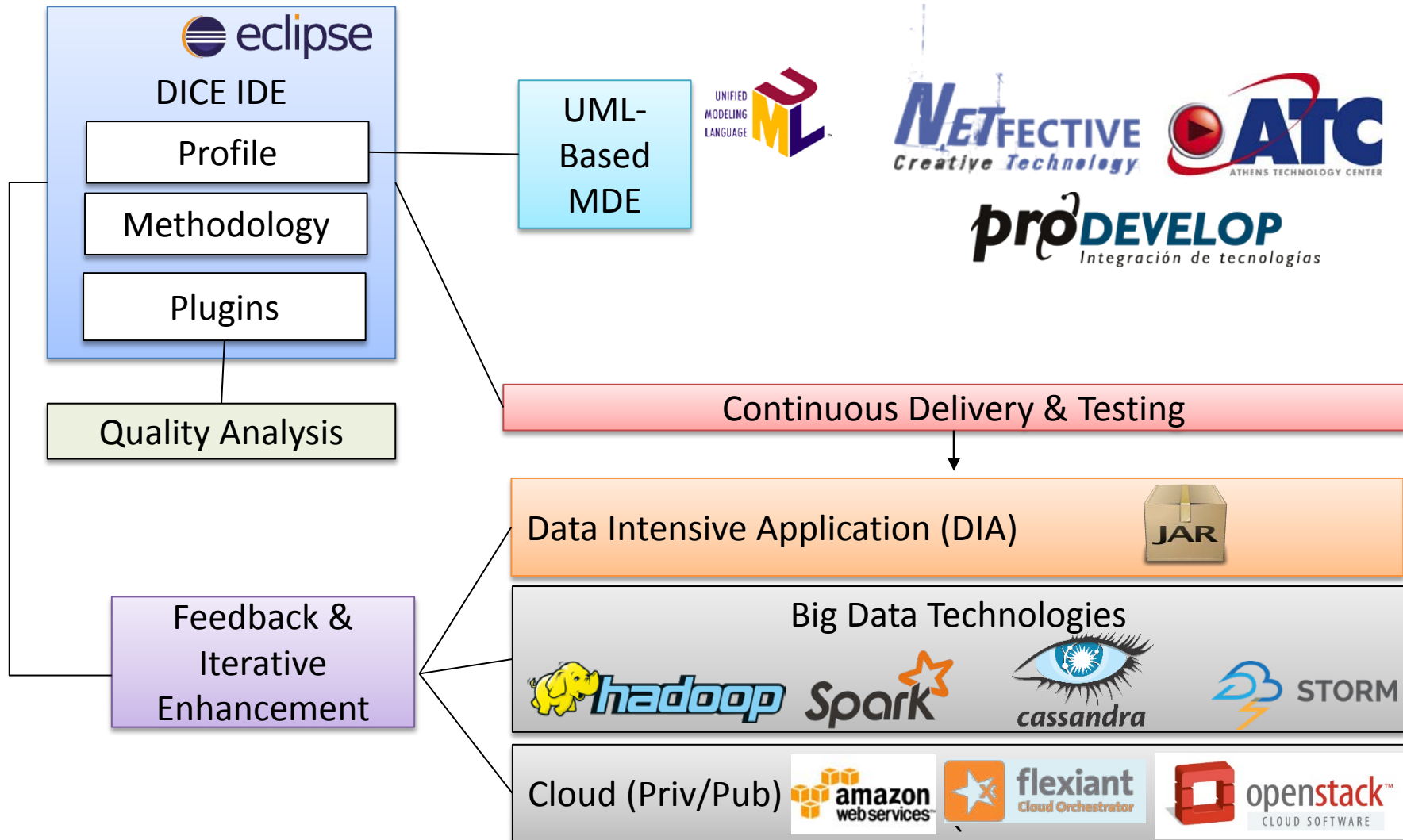
- Performance
- Costs

○ Correctness

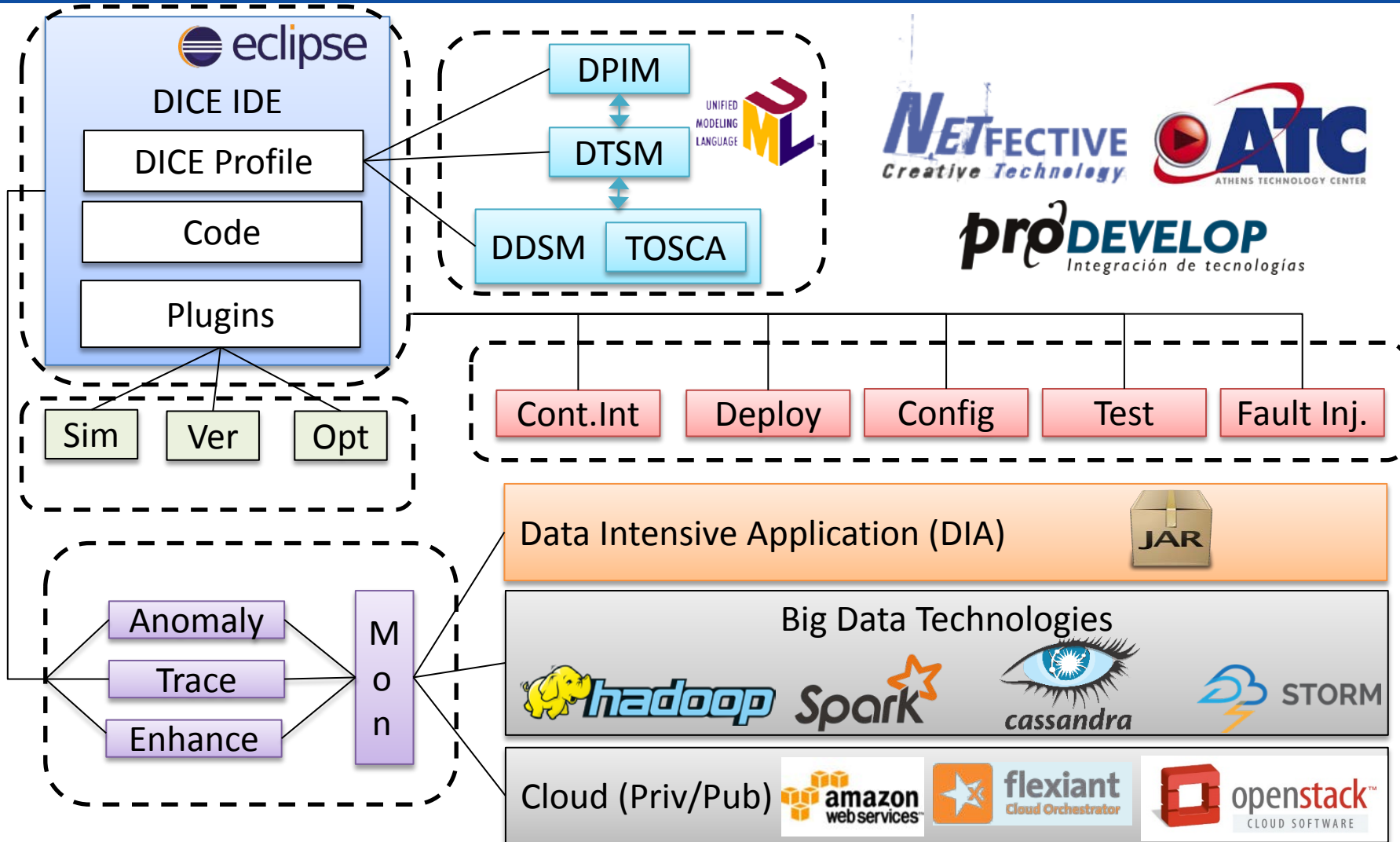
- Privacy & security
- Temporal metrics



# DICE Framework



# DICE Framework



# DICE Workflow - Dev



DICE Eclipse IDE



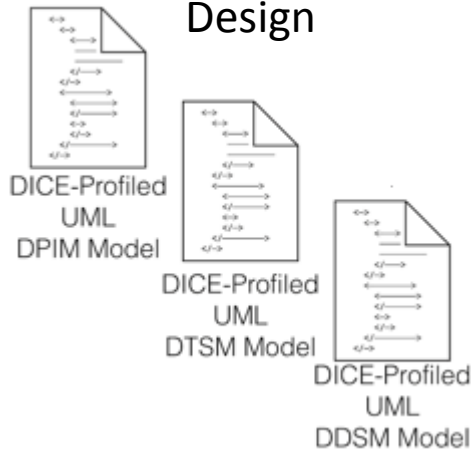
“I want to design for Big data”



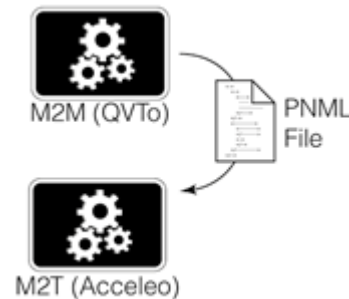
“Will the DIA meet SLAs and costs?”

Enhance

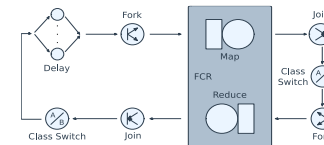
Design



Transform to Formal Models



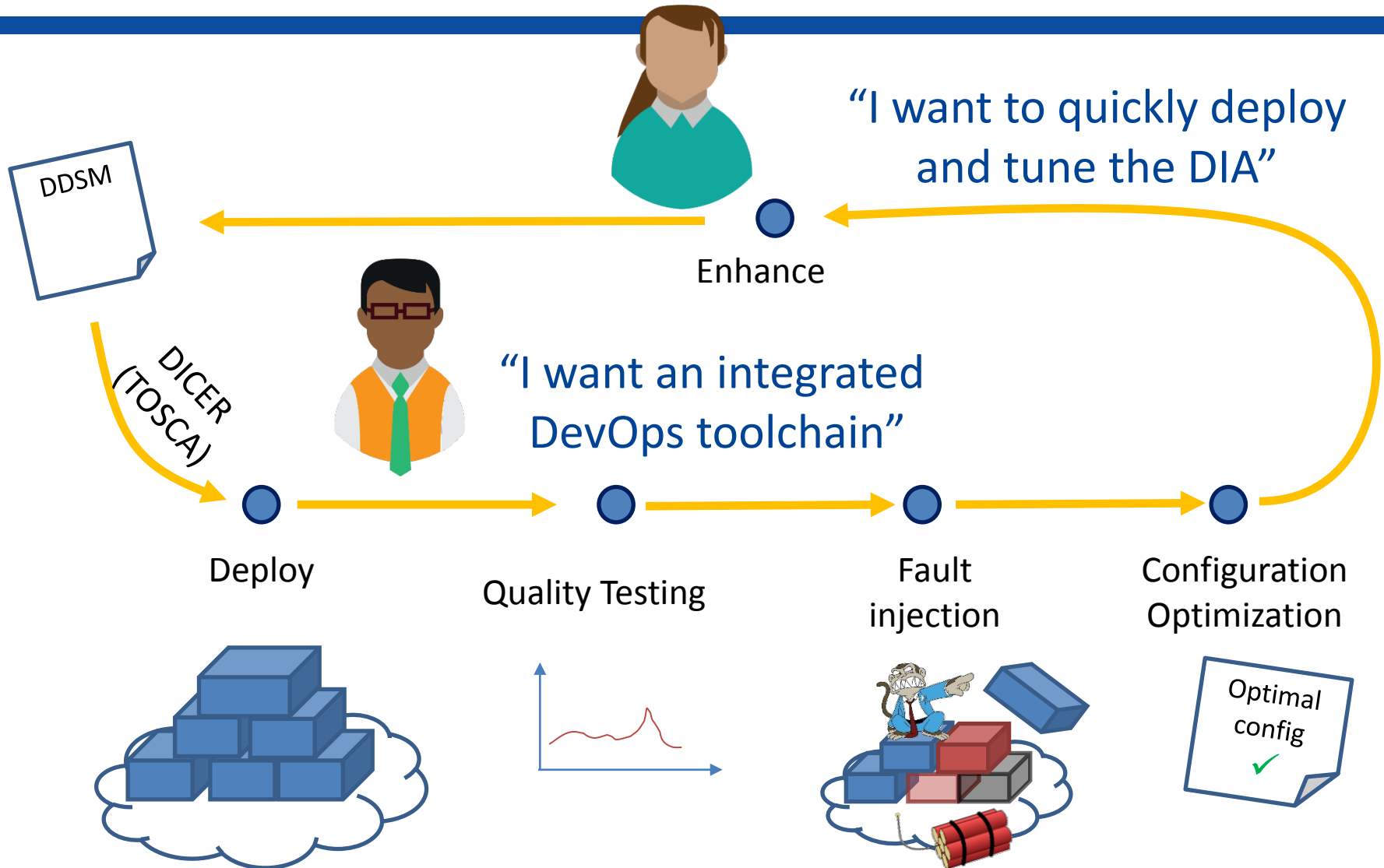
Simulate & Verify



Optimize



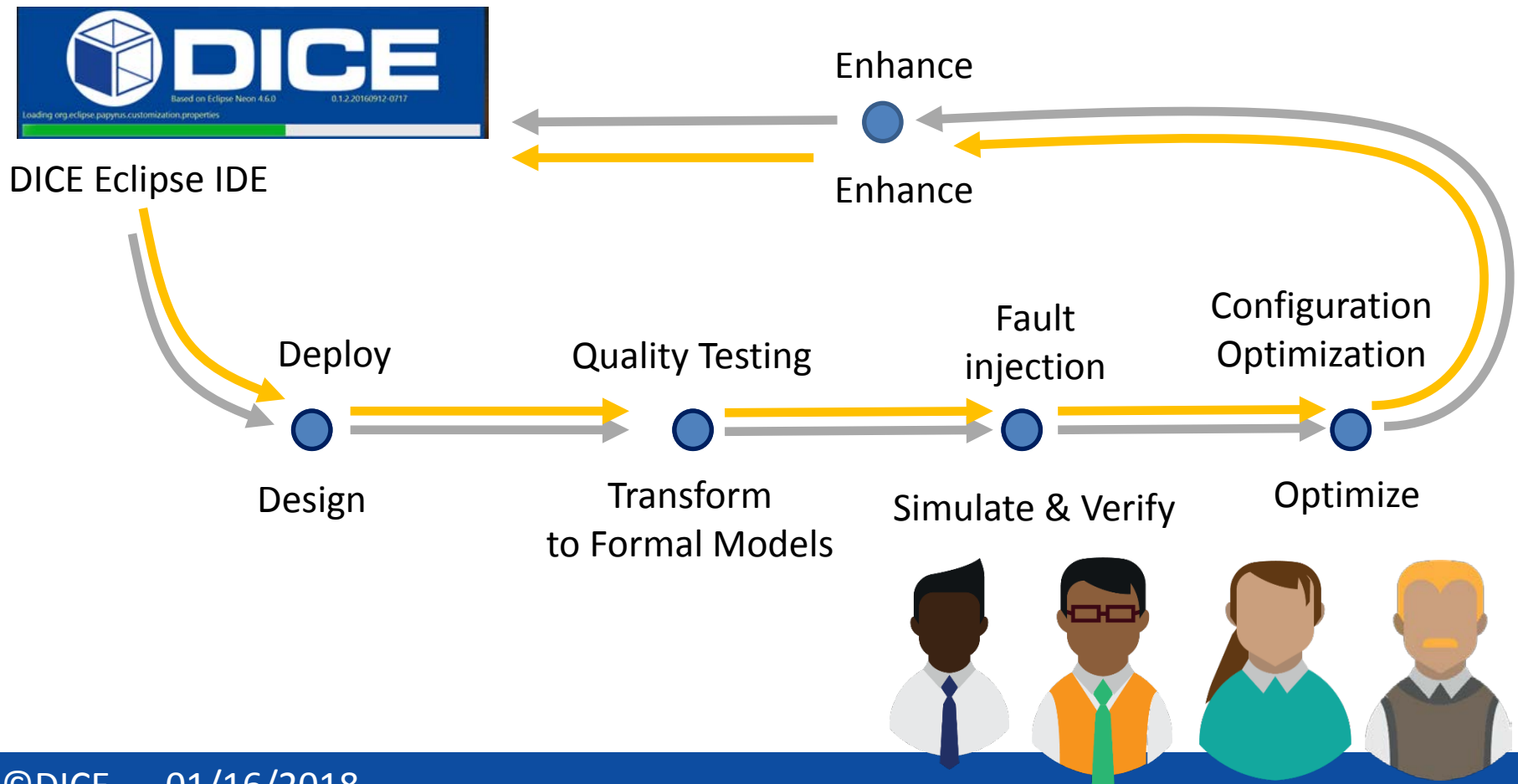
# DICE Workflow - Ops



# DICE Workflow: Unified Toolchain



Following the DevOps paradigm, DICE delivers a unified toolchain for the enterprise team





# DICE Approach, Tools, Architecture and Methodology

DICE Horizon 2020 Project  
Grant Agreement no. 644869  
<http://www.dice-h2020.eu>

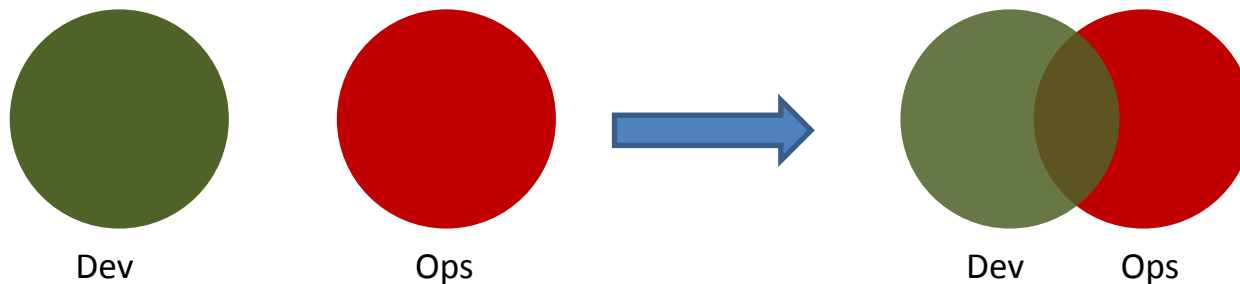


Funded by the Horizon 2020  
Framework Programme of the European Union

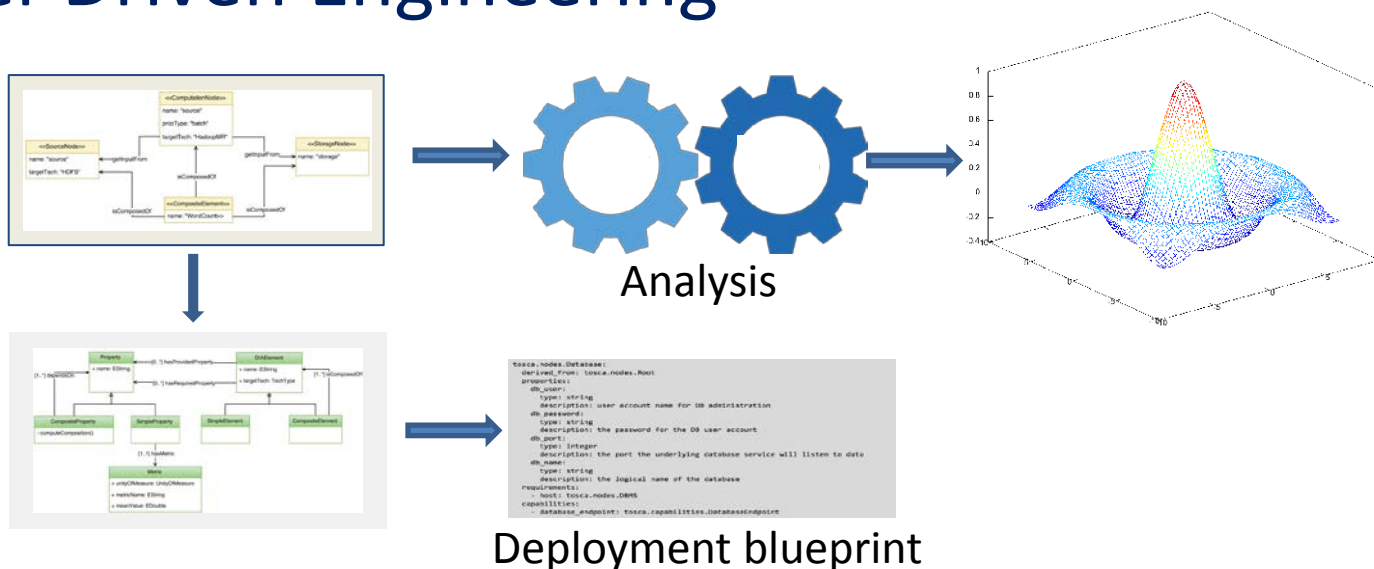
# Ingredients of technical approach



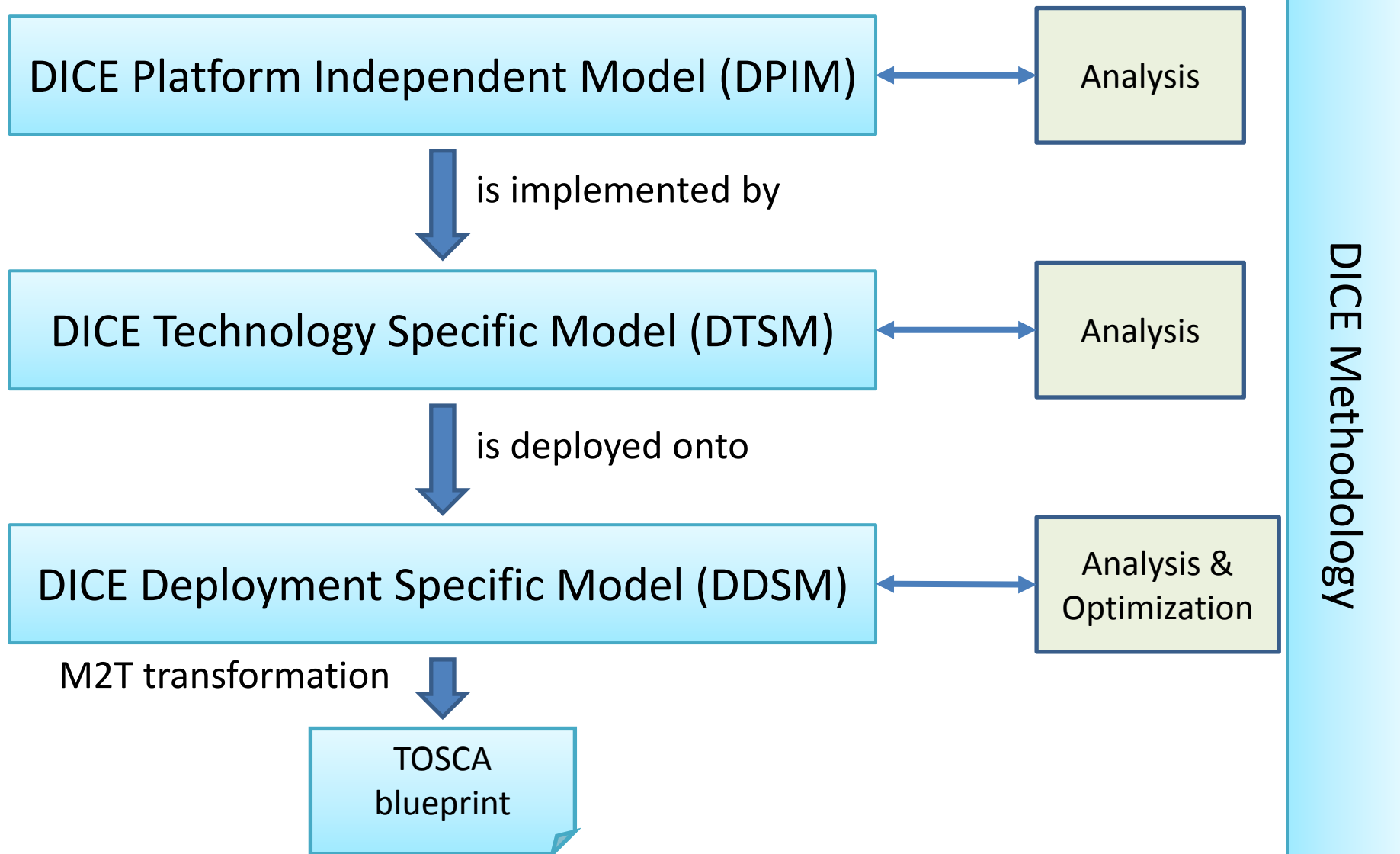
## o DevOps



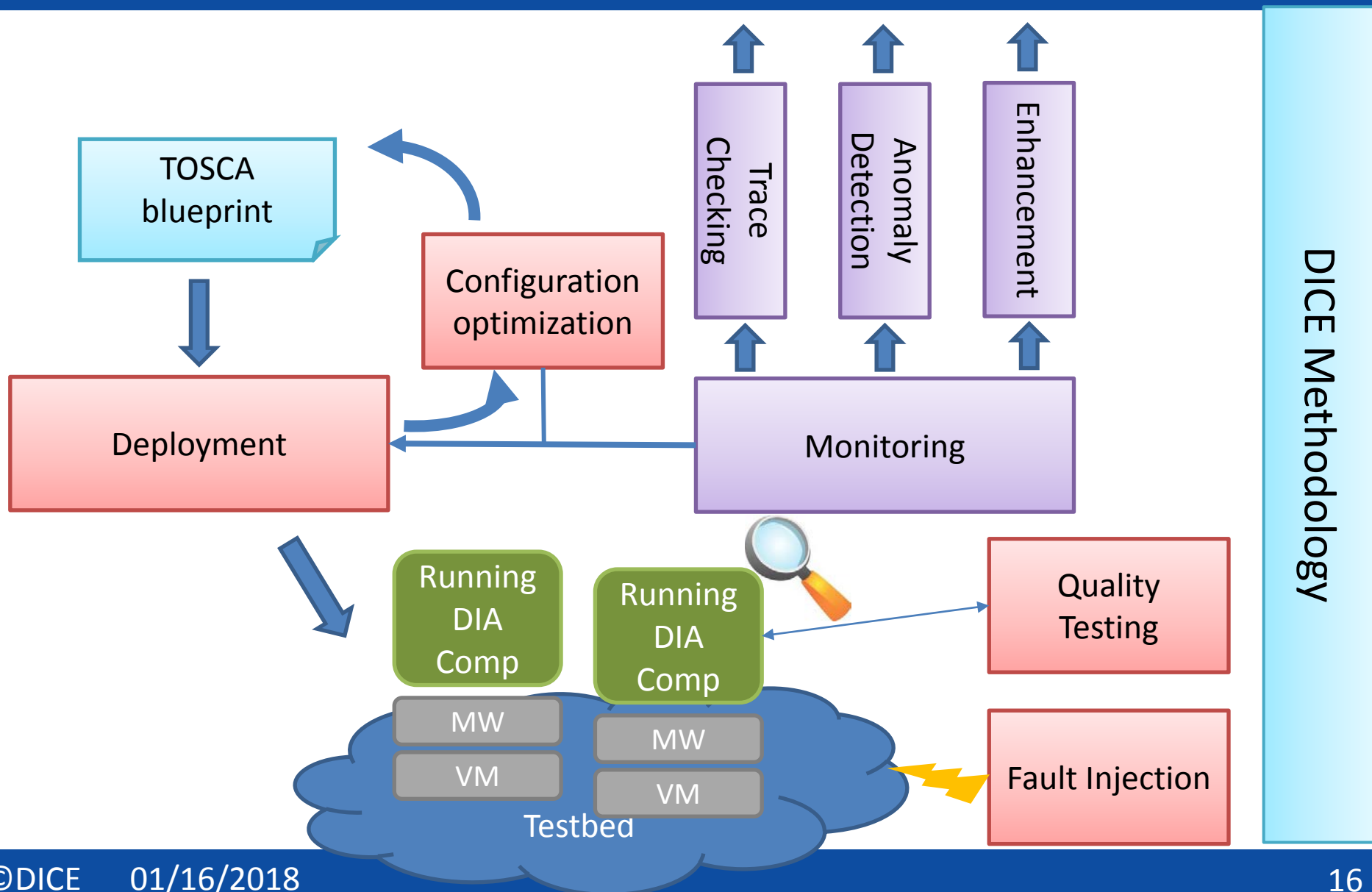
## o Model-Driven Engineering



# DICE incremental modeling and analysis



# DICE deployment, monitoring and testing



# Modeling in the IDE



The screenshot shows the DICE Platform IDE interface. The main workspace displays a DPIM Activity Diagram for 'WikistatsDPIM'. The diagram includes a start node, a '«GaWorkloadEvent»' element with an 'arrivalRate=5.0' note, a 'Partition1' container with a '«GaStep» WikistatsApplication' element, and an end node. Notes specify 'usedResources=cluster1 resp1=5.46' and 'rep=1 prob=1 hostDemand={2,0}'. The right sidebar contains a task list:

- Apply MARTE and DICE profiles to the model
- Create 2 diagrams to the model
- Complete the Deployment diagram
- Complete the Activity diagram

The Activity diagram should be completed adding some elements

- Add an "Initial Node" element inside the Activity
- Add an "Activity Partition" element inside the Activity. Open the Properties view, and select the UML tab and set the property "Represents" to the Artifact created in the previous step. Use the "3 points" button and browse the tree model until you found the desired element
- Add an "Opaque Action" element inside the Activity Partition
- Add an "Activity Final Node" element inside the Activity. This element is placed in the same entry of the Initial Node in the Palette
- Add a "Control flow" link between the "Initial Node" and the "Opaque Action"
- Add a "Control flow" link between the "Opaque Action" and the "Activity Final Node"

Additional tasks:

- Stereotype the Deployment elements
- Stereotype the Activity elements
- Configure the launching
- See the results of the launching

At the bottom, a navigation bar shows 'OSM', 'DPIM Class Diagram', and 'DPIM Activity Diagram'. A bottom right panel lists 'GaScenario (Structured)', 'MARTE/HLAM', and 'MARTE/GCM'. A small avatar icon is visible in the bottom left corner.

I'm defining the DPIM model for my application

# DPIM-level simulation



A two VMs configuration will not be acceptable for this application

Name: Wikistats-1

Main Filters Parameters Advanced Common

Model to Analyse

platform:/resource/modelForReliabilityTest/wikistatsscreenshots/wikistats.uml Browse

Active scenario

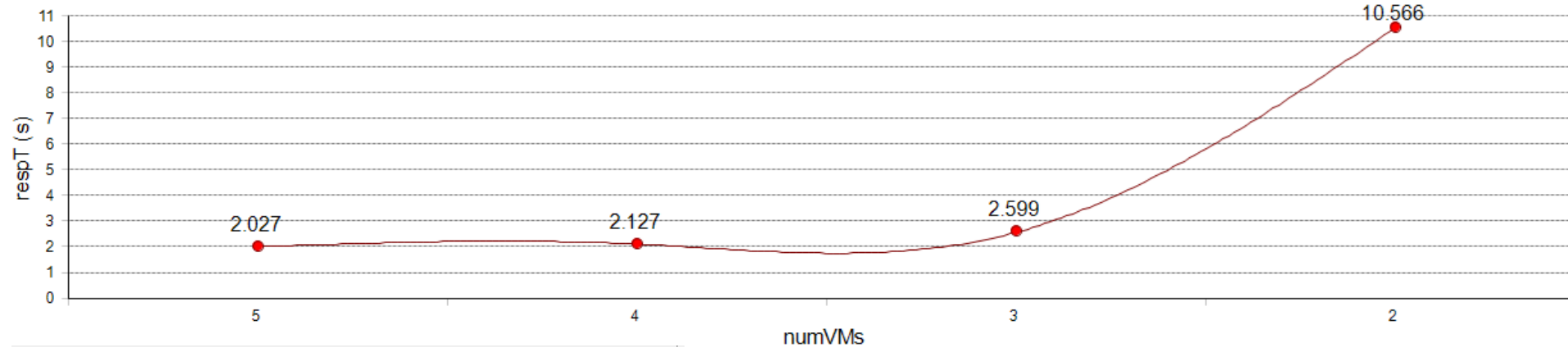
- <<StormScenarioTopology, GaAnalysisContext>> <Activity> WikistatsTopology
- <<StormScenarioTopology>> <Class> WikistatsStormTopology
- <<DpimScenario, GaAnalysisContext>> <Activity> WikistatsDPIM
- <<HadoopScenario, GaAnalysisContext>> <Activity> WikistatsBatchJob

NFP to calculate

Performance

Reliability

9d02fb7d-4660-4bd0-bf42-b67feb3d6224



Run

Close



# The optimization step

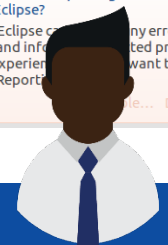


The screenshot shows the Eclipse IDE interface. The main editor displays a MARS (Model-Driven Architecture Scenario) diagram titled "Topology" with a "Partition 1" container. Inside the partition, there is a flow from an initial node to a node labeled "«StormSpout» WikiArticlesSp...", which then flows to a node labeled "«StormStreamStep» «StormBolt» LinkCounter...", and finally to a "Final" node. The Package Explorer on the left shows a project named "Wikistats" with several sub-projects. The Class page dialog is open in the foreground, showing a list of VM configurations: "Amazon-large", "Amazon-xlarge", "Amazon-2xlarge", and "Cineca-5xlarge". The "Amazon-large" configuration is selected. The dialog also shows "Amazon-medium" as an alternative. There are buttons for "Load DDSM for this class...", "Refresh vm configurations", and navigation buttons "< Back" and "Next >".

Max>> Everything seems to be ok. Let's focus on optimization...

Let's identify the optimal configuration for Amazon reserved instances

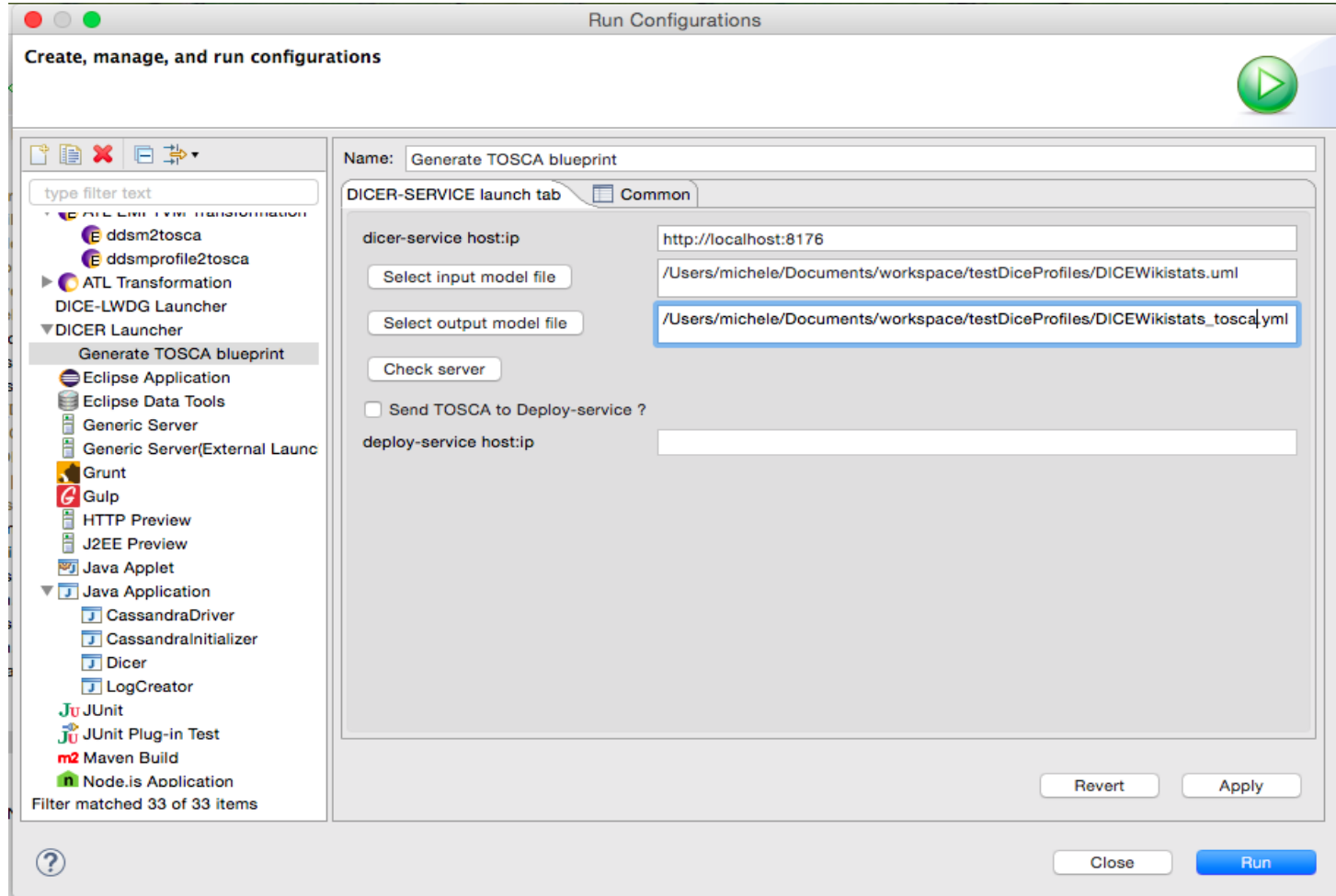
(Waiting for Focus) Eclipse Error Reporting  
Welcome to the Eclipse Error Reporting Service. Do you want to help Eclipse? With your permission Eclipse can log any errors logged inside the IDE and inform you about the issues you experienced. You can want to help out by enabling Error Reporting. [Yes] [No] [Disable]



# Deployment step– creating the TOSCA YAML



I have got all I need to create the blueprint!



# Deployment step – sending the TOSCA YAML to the deployment service



... and now let's  
deploy!

Run Configurations

Create, manage, and run configurations

Invalid resources path.

Name: wikistats

Deployment details

Blueprint

Main blueprint file: /Users/elisabettadinitto/Documents/DICEWorkspace/DICE-WikiStats/model/ Browse

Resources folder: Browse

Deployment Service

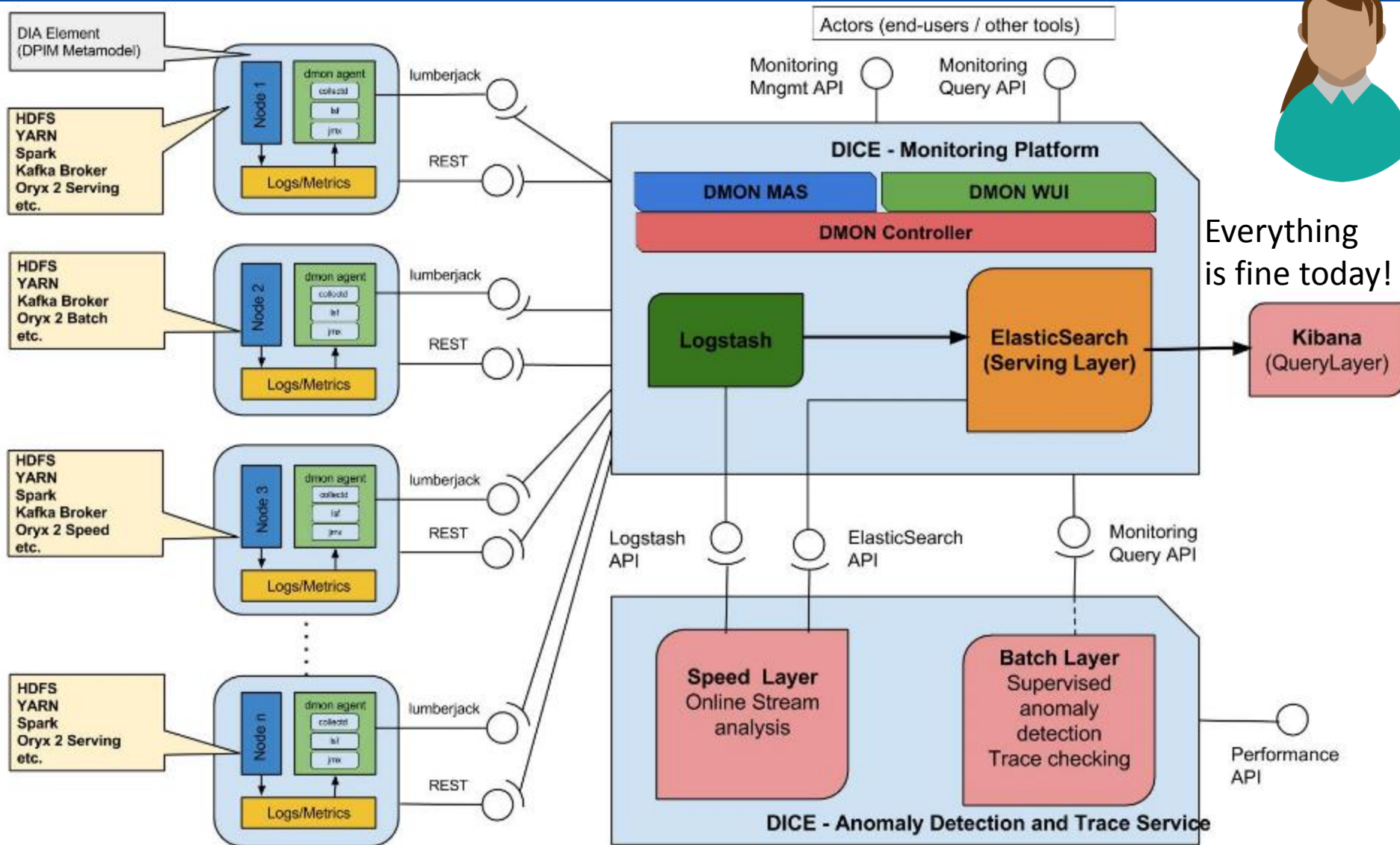
Deployment Service: fco (http://109.231.122.194/) ⌵

Container: WikiStats (e65e69ab-fb28-4ac2-b808-6c725fb08b93) ⌵

Revert Apply

Close Run

# Monitoring and detecting anomalies



Everything is fine today!

Kibana (QueryLayer)

Performance API

DICE - Anomaly Detection and Trace Service

# Quality testing



## Features:

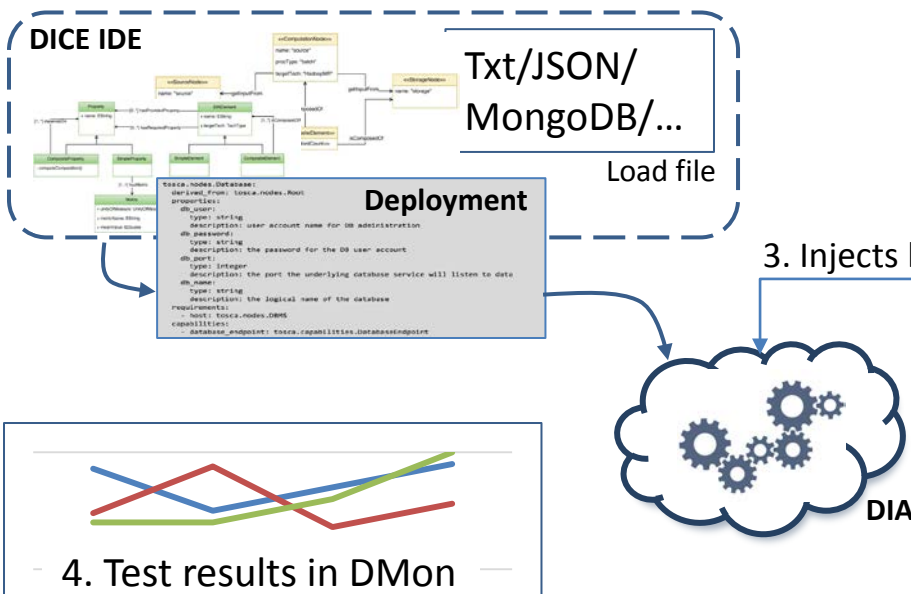
- Load generation for stream processing systems
- Load replay
- Load scaling (via Hidden Markov Model representation)



Load injection  
into the DIA

## QT Tool workflow:

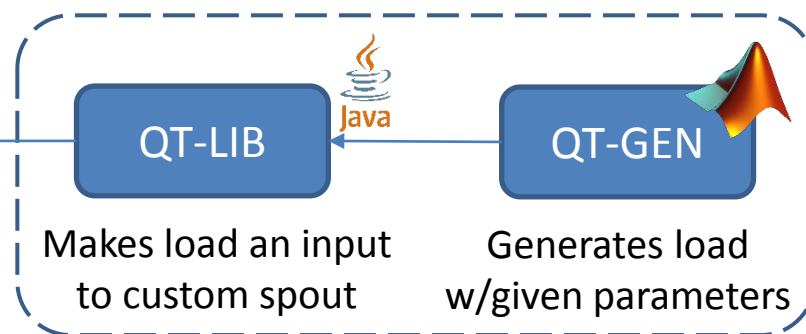
1. Max & Ophra add QT tool as a spout, provide (initial) load & deploys DIA



Let's stress our  
DIA!



2. QT tool starts simultaneously with DIA



3. Injects load



4. Test results in DMon

# Fault injection

flexiant™ Cloud Orchestrator Admin

Clusters  
Nodes  
Networks

DICE Fault Injection Tool

Discover Visualize Dashboard Settings

### DICE FIT High CPU

IP Address \*

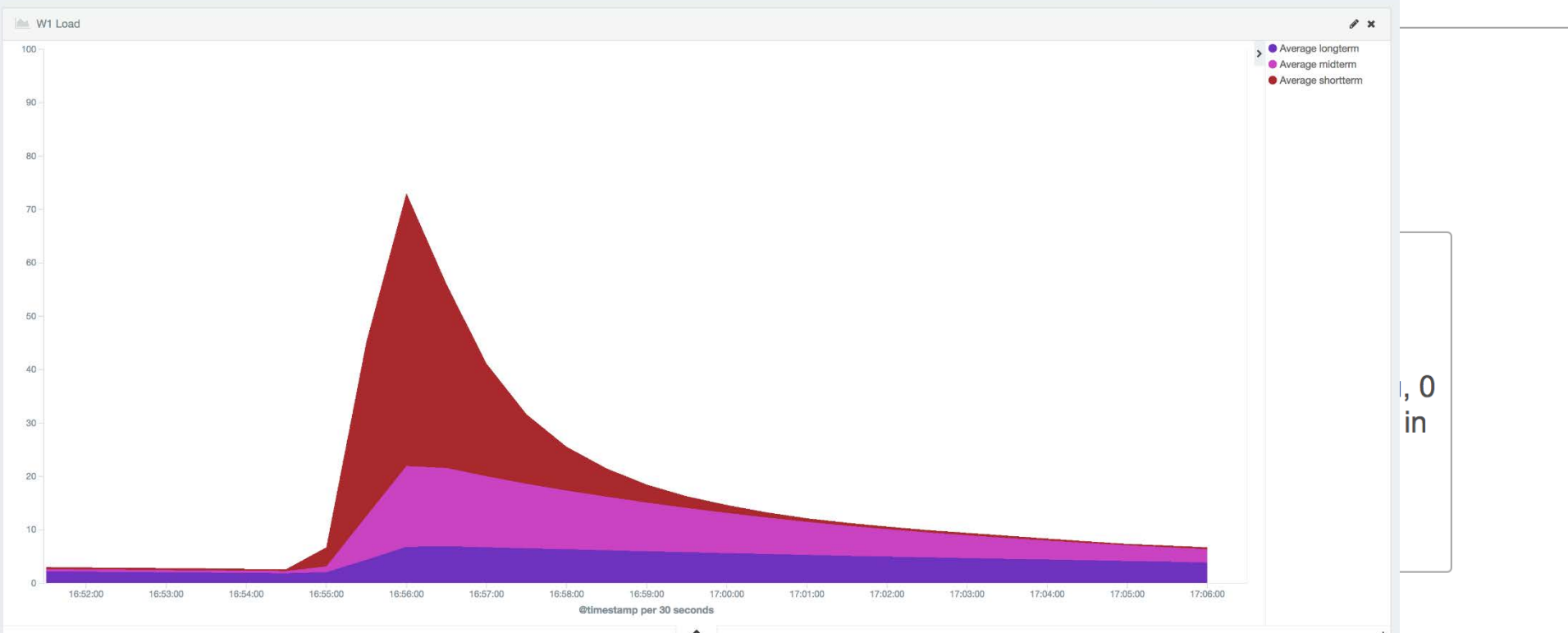
Username \*

Password \*

CPU count \*

Time \*

Last 15 minutes



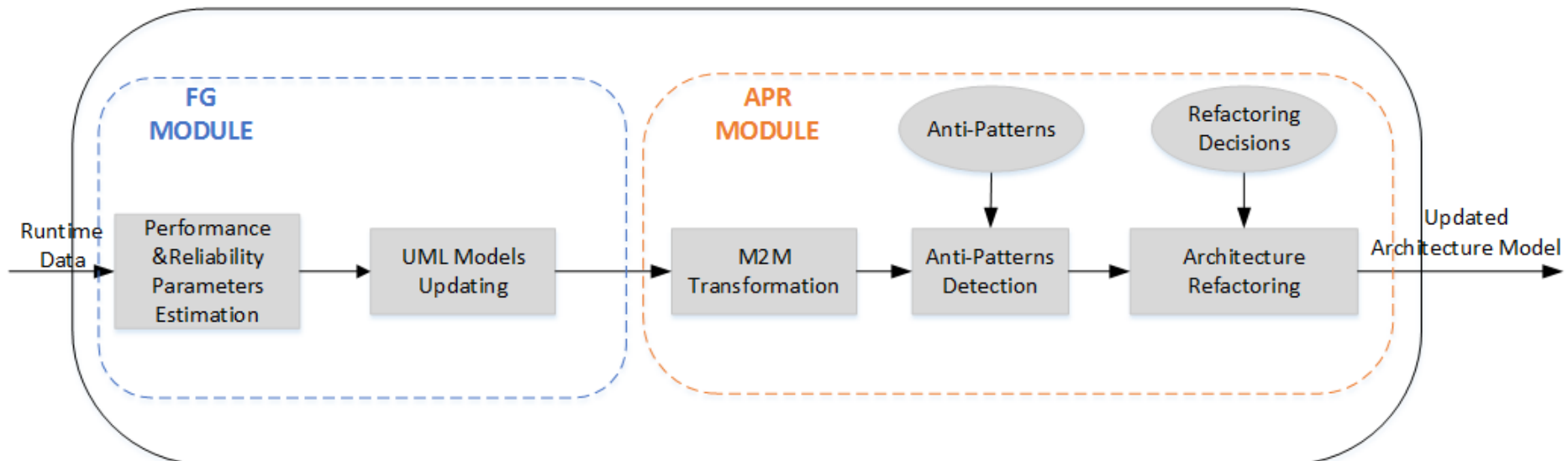
, 0  
in

# Enhancement tool



- Objective: designed for iteratively enhancing the DIA quality
- Functions:
  - Providing a performance and reliability analysis
  - Updating UML models with analysis results
  - Anti-patterns Detection
  - Refactoring the design model

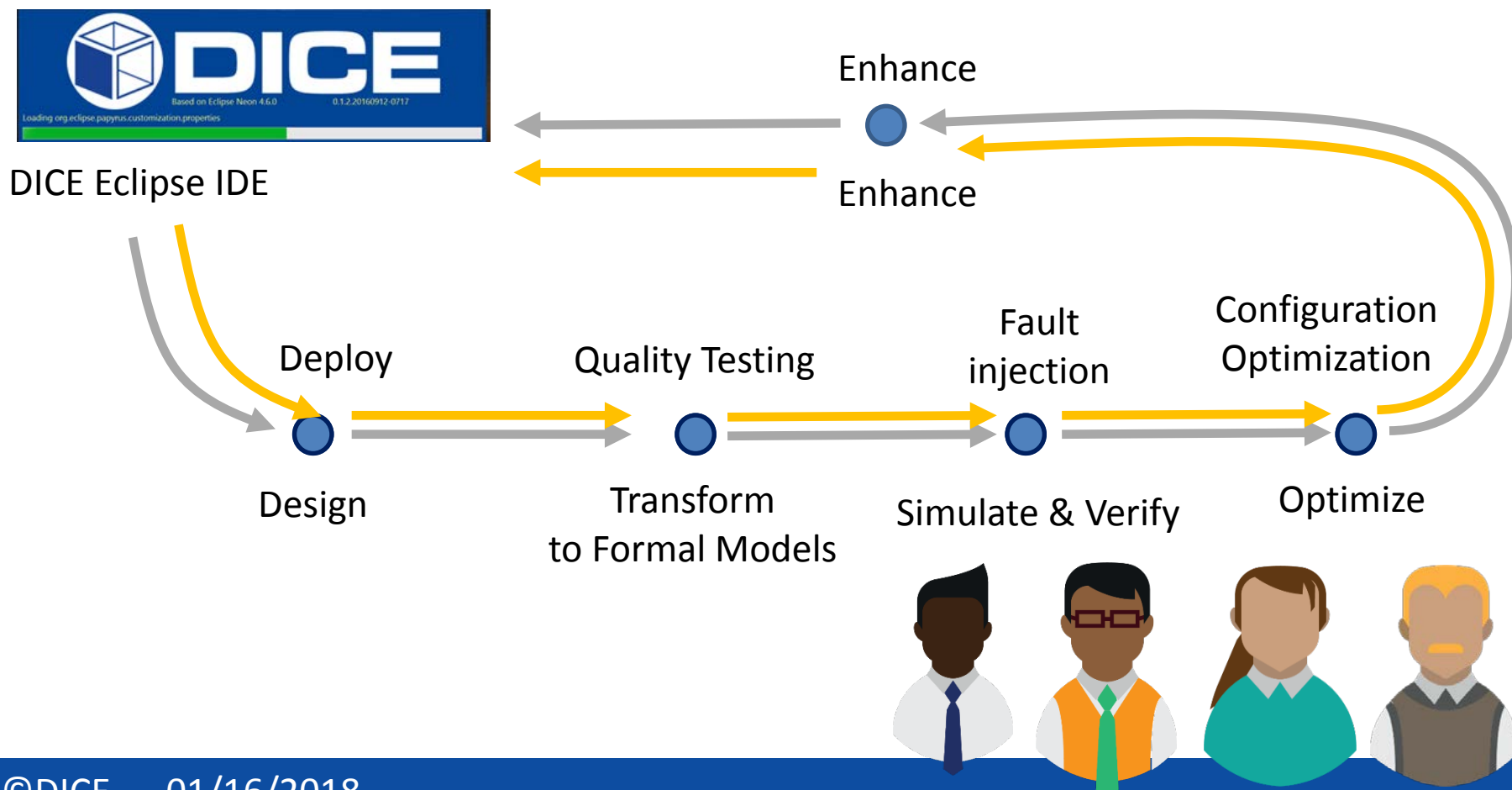
let's consider the proposed improvements!



# Summing up



Following the DevOps paradigm, DICE delivers a unified toolchain for the enterprise team



# Demonstrators



News Asset



News&Media  
Market



Tax Fraud Detection  
Application



e-Government  
Market

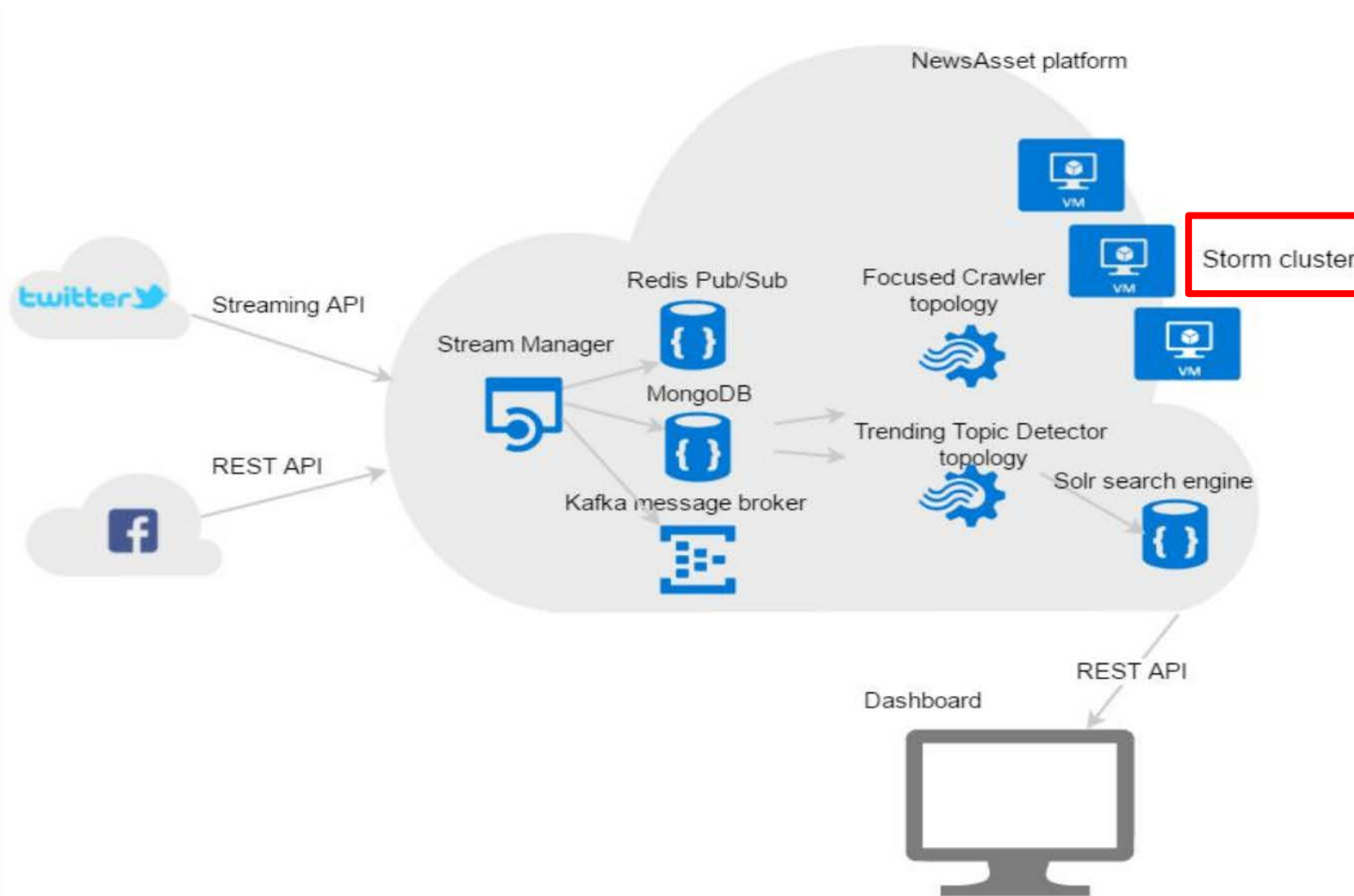


Posidonia Operations



Maritime  
Sector

# News analysis demonstrator



# Configuring a Big data system



```
102
103 drpc.port: 3772
104 drpc.worker.threads: 64
105 drpc.max_buffer_size: 1048576
106 drpc.queue.size: 128
107 drpc.invocations.port: 3773
108 drpc.invocations.threads: 64
109 drpc.request.timeout.secs: 600
110 drpc.childopts: "-Xmx768m"
111 drpc.http.port: 3774
112 drpc.https.port: -1
113 drpc.https.keystore.password: ""
114 drpc.https.keystore.type: "JKS"
115 drpc.http.creds.plugin: org.apache.storm.security.auth.DefaultHttpCredentialsPlugin
116 drpc.authorizer.acl.filename: "drpc-auth-acl.yaml"
117 drpc.authorizer.acl.strict: false
118
119 transactional.zookeeper.root: "/transactional"
120 transactional.zookeeper.servers: null
121 transactional.zookeeper.port: null
122
123 ## blobstore configs
124 supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
125 supervisor.blobstore.download.thread.count: 5
126 supervisor.blobstore.download.max_retries: 3
127 supervisor.localizer.cache.target.size.mb: 10240
128 supervisor.localizer.cleanup.interval.ms: 600000
129
```



# DICE configuration optimization

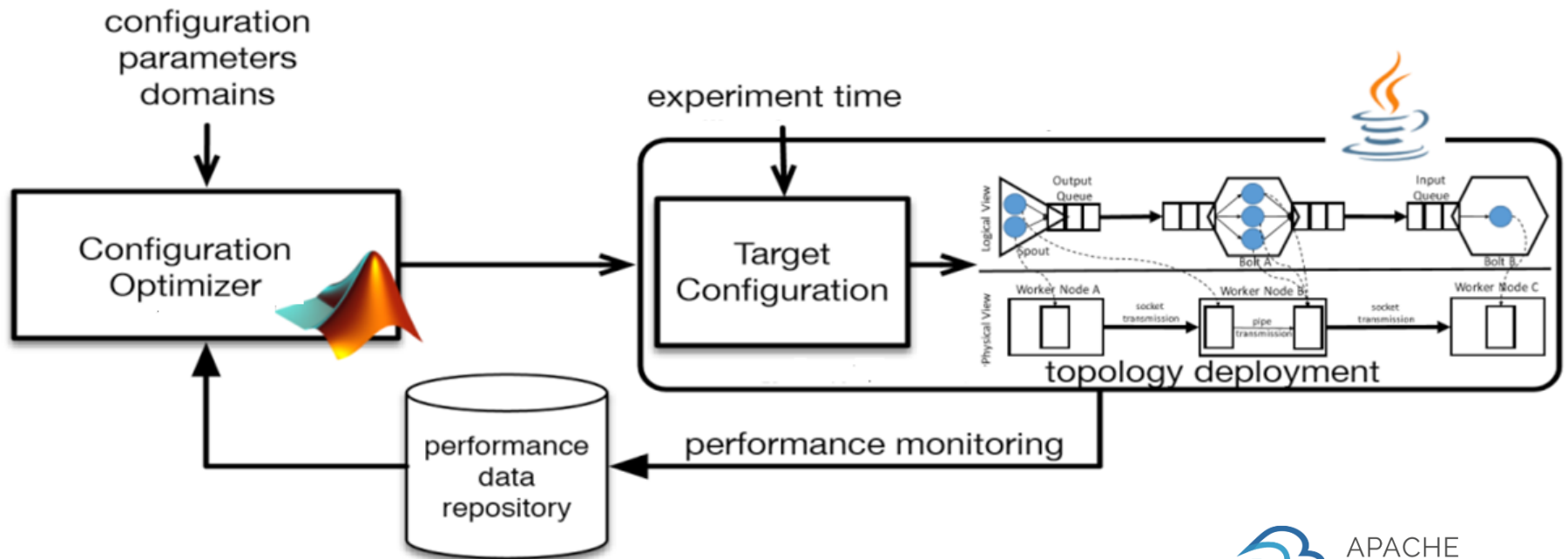


The screenshot displays the DICE Configuration View interface. On the left, a Storm topology diagram is shown, divided into Partition 1 and Partition 2. Partition 1 contains a spout (spout\_1) and a bolt (bolt\_1). Partition 2 contains a spout (spout\_2) and a bolt (bolt\_2). The diagram shows data flow between these components via StormStreamStep and Bolt objects (BJ1, BM1). Each component has associated configuration parameters such as parallelism, hostDemand, numTuples, and grouping.

On the right, the 'DICE Configuration View' window is open, showing a list of parameters for the 'storm' plugin. The 'Parameter Selection' tab is active. Below the list, there is an 'Add Parameters' section with a table of parameters.

Parameter	Type	Min	Max	Step	Options
topology.error.throttle.interval.secs					
topology.trident.batch.emit.interval.millis					
topology.disruptor.wait.timeout.millis					
topology.disruptor.batch.size					
topology.disruptor.batch.timeout.millis					
topology.disable.loadaware.messaging					
topology.state.checkpoint.interval.ms					
topology.max.spout.pending					
topology.acker.executors					
topology.tick.tuple.freq.secs					
<b>Add Parameters</b>					
topology.executor.receive.buffer.size	Integer	1024	2048	1	
topology.min.replication.count	Integer	1	10	1	
topology.worker.shared.thread.pool.size	Integer	1	20	1	
topology.max.task.parallelism	Integer	1	50	1	

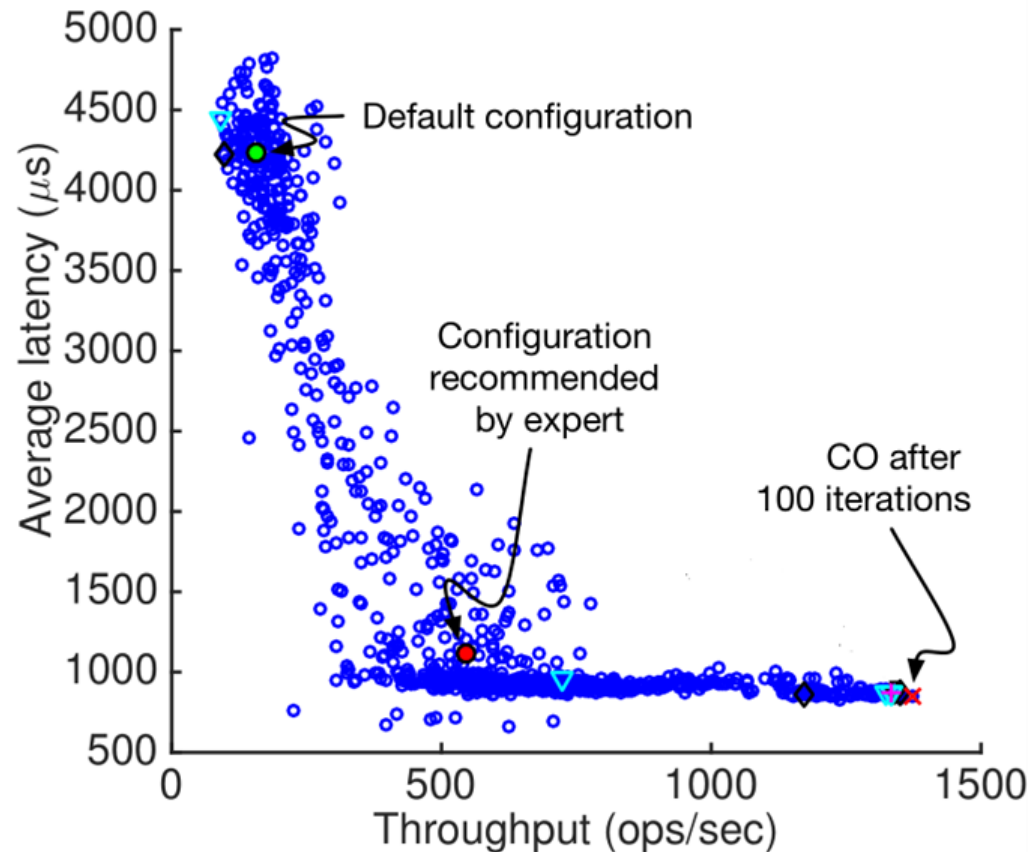
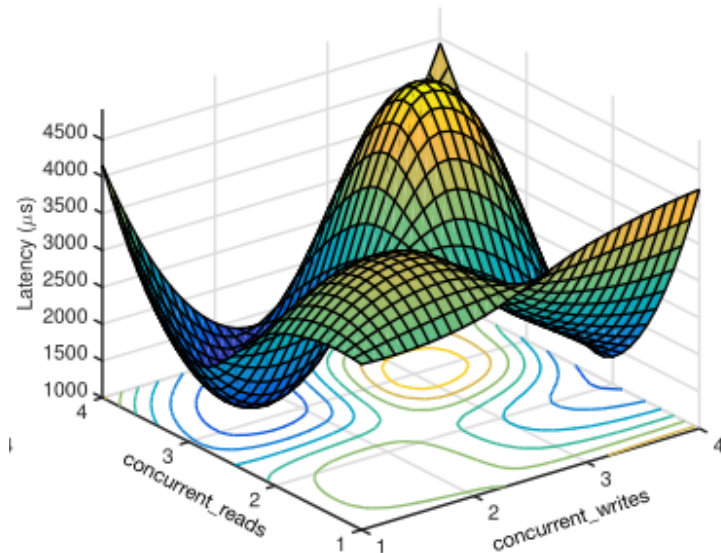
# DICE configuration optimization



# DICE configuration optimization



Applying Bayesian  
Optimization to DevOps



# Thanks!



WWW.DICE-H2020.EU

DEVOPS  
FOR  
BIG DATA

**DICE** Developing Data-Intensive Cloud Applications with Iterative Quality Enhancements

Tools

A high-level overview of the DICE framework is shown in the figure below. The framework includes the following components:

- DICE IDE**: an integrated development environment for iterative coding, design and application prototyping, based on the Eclipse IDE and the TICE (IDE) framework.
- Quality analysis tools**: a set of tools for quality analysis during the early stages of application design via simulation, verification and validation methods.
- Feedback and Iterative Enhancement tools**: a monitoring platform tailored to big data technologies and coupled with tools that will enable an iterative quality enhancement of different application designs and explore how the data intensive application evolves over time.
- Continuous Delivery and Testing tools**: a set of tools and methods supporting delivery of private and public clouds via a TOGAF compliant layered model, central application configuration, continuous integration and quality testing.

Diagram components: Eclipse IDE, DICE IDE, Profile, Methodology, Plugins, Quality Analysis, UML-based IDE, Neo4j, ATC, proDEVELOP, Continuous Delivery & Testing, Data Intensive Application (DIA), Big Data Technologies.