

DICE: Quality-Aware DevOps For Big Data Applications

GIULIANO CASALE, IMPERIAL COLLEGE LONDON
(On behalf of the DICE consortium)

G.CASALE@IMPERIAL.AC.UK



DICE Research Project

Horizon 2020 research project (4M€, 2015–18)



Consortium:



Imperial College
London



Universidad
Zaragoza



POLITECNICO
DI MILANO

pro²DEVELOP
Integración de tecnologías

flexiOPS



Our mission: supporting SMEs in developing high-quality Big data applications



**“...AND AFTER BIG DATA SOLVES ALL OUR PROBLEMS,
WE’LL RIDE AWAY ON MAGIC FLYING UNICORNS”**



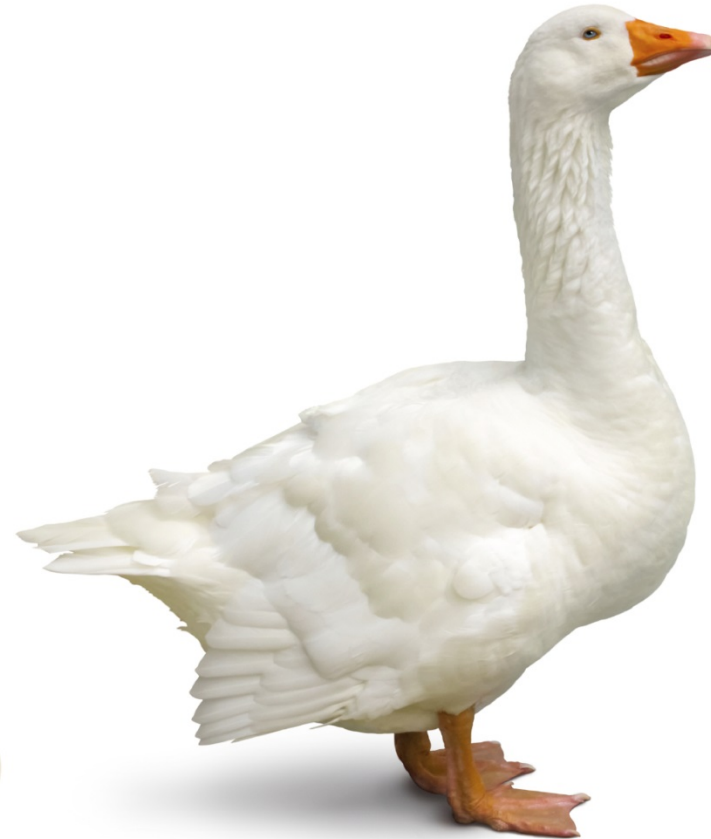
A story from our consortium...

Company
core market:
Legacy software
modernization



A story from our consortium...

New market:
Analysis of legacy
Big data



The barrier: a steep learning curve!



www.StrangeTravel.com

Example: 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
```



Riddles for the brave

- Will my Big data applications
- How do I select the right Big technology?
- How do I choose the best architecture? And how to verify that I am right?
- What if service-level agreements change?

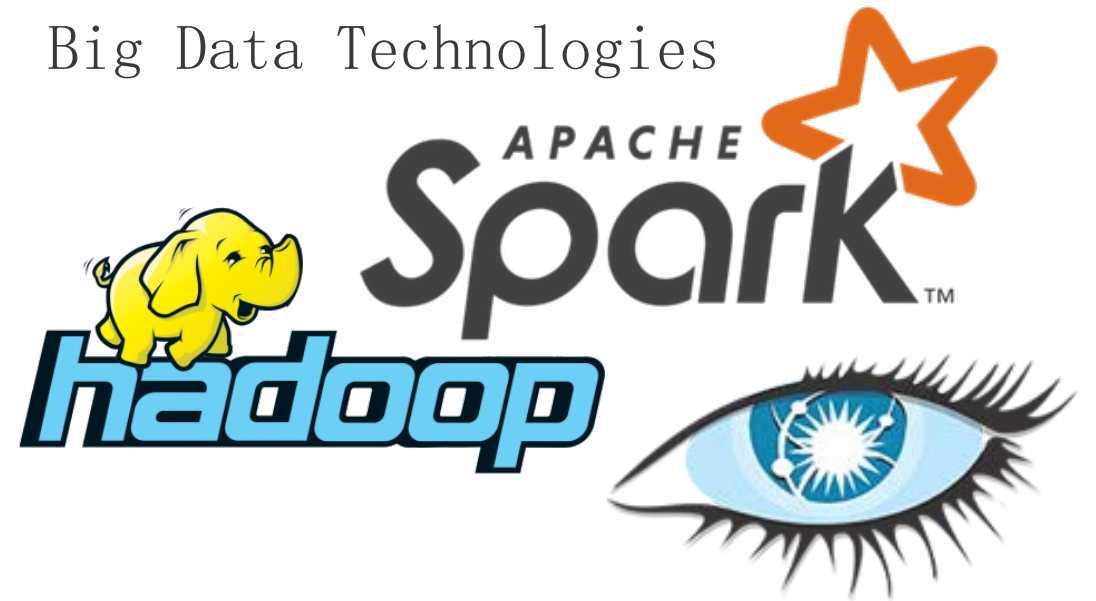


Our Concerns

Characterize Data Properties



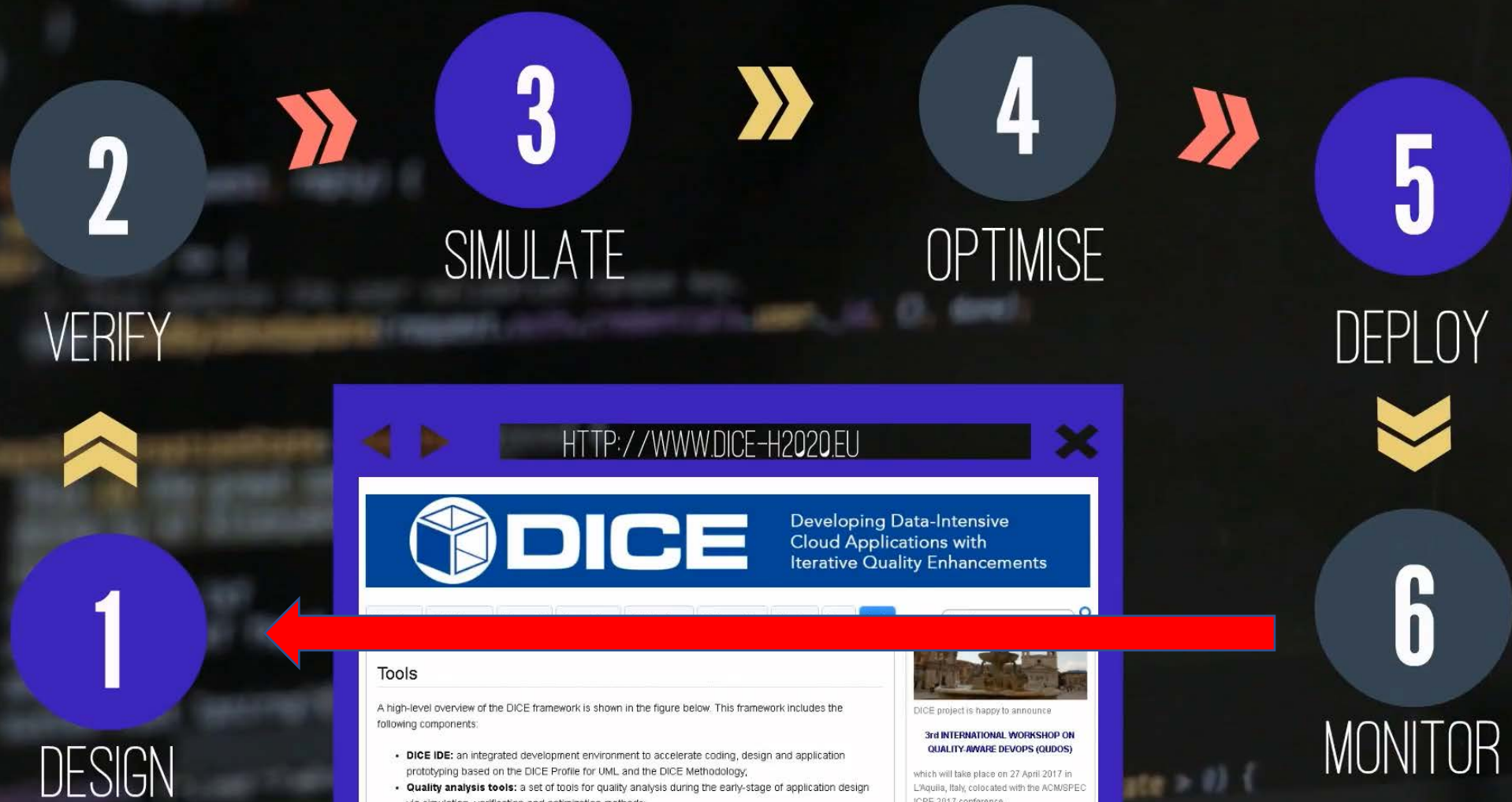
DevOps tools for popular
Big Data Technologies



Quality-Aware
Reasoning Tools



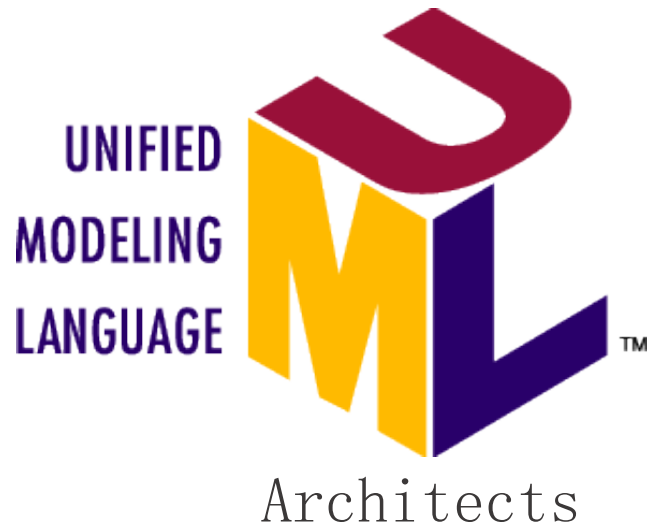
DICE: a DevOps toolchain for Big data applications



Different perspectives



Developers



Ops



Testers

The problem with creating a unified view

PERIODIC TABLE OF DEVOPS TOOLS (V2)

1 Fm Gh Github																			2 Fm Aws AmazonWeb Services						
3 Os Gt Git	4 En Dm DBmaestro																			5 En Ch Chef	6 En Pu Puppet	7 Os An Ansible	8 Os Sl Salt	9 Os Dk Docker	10 Pd Az Azure
11 Fm Bb Bitbucket	12 Os Lb Liquibase																			13 Os Ot Otto	14 En Bl BladeLogic	15 Os Va Vagrant	16 Fr Tf Terraform	17 Os Rk rkt	18 En Gc Google Cloud Platform
19 Os Gl GitLab	20 En Rg Redgate	21 Os Mv Maven	22 Os Gr Gradle	23 Os At ANT	24 Os Fn FitNesse	25 Fr Se Selenium	26 Os Ga Gatling	27 Fr Dh Docker Hub	28 Os Jn Jenkins	29 Pd Ba Bamboo	30 Os Tr Travis CI	31 Pd Gd Deployment Manager	32 Os Sf SmartFrog	33 Os Cn Consul	34 Os Bc Bcfg2	35 Os Mo Mesos	36 En Rs Rackspace								
37 Os Sv Subversion	38 En Dt Datical	39 Os Gt Grunt	40 Os Gp Gulp	41 Os Br Broccoli	42 Fr Cu Cucumber	43 Os Cj Cucumber.js	44 Fr Qu Qunit	45 Os Npm npm	46 Fm Cs Codeship	47 Pd Vs Visual Studio	48 Fm Cr CircleCI	49 Fr Cp Capistrano	50 Fr Ju JuJu	51 Os Rd Rundeck	52 Os Cf CFEngine	53 Fr Ds Swarm	54 Os Op OpenStack								
55 Os Hg Mercurial	56 En Dp Delphix	57 Fr Sb sbt	58 Os Mk Make	59 Os Ck CMake	60 Fr Jt JUnit	61 Fr Jm JMeter	62 Fr Tn TestNG	63 Os Ay Artifactory	64 Fm Tc TeamCity	65 Fm Sh Shippable	66 Os Cc CruiseControl	67 En Ry RapidDeploy	68 Fm Cy CodeDeploy	69 En Oc Octopus Deploy	70 En No CA Nolio	71 Os Kb Kubernetes	72 Fm Hr Heroku								
73 En Cw ISPW	74 En Id Idera	75 Os Msb MSBuild	76 Os Rk Rake	77 Fr Pk Packer	78 Os Mc Mocha	79 Fr Km Karma	80 Os Jm Jasmine	81 Os Nx Nexus	82 Os Co Continuum	83 Fm Ca Continua CI	84 Pd So Solano CI	85 En Xld XL Deploy	86 En Eb ElasticBox	87 Fm Dp Deploybot	88 En Ud UrbanCode Deploy	89 Os Nm Nomad	90 En Os OpenShift								

PERIODIC TABLE OF DEVOPS TOOLS (V2)

- Os Open Source
- Fr Free
- Fm Freemium
- Pd Paid
- En Enterprise
- SCM
- CI
- Deployment
- Cloud / IaaS / PaaS
- BI / Monitoring
- Database Mgmt
- Repo Mgmt
- Config / Provisioning
- Release Mgmt
- Logging
- Build
- Testing
- Containerization
- Collaboration
- Security

91 En Xlr XL Release	92 En Ur UrbanCode Release	93 En Bm BMC Release Process	94 En Hp HP Codar	95 En Au Automic	96 En Pl Plutora Release	97 En Sr Serena Release	98 Pd Tfs Team Foundation	99 Fm Tr Trello	100 Pd Jr Jira	101 Fm Rf HipChat	102 Fm Sl Slack	103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker	105 En Sn ServiceNow
106 Os Ki Kibana	107 Fm Nr New Relic	108 En Dt Dynatrace	109 Os Ni Nagios	110 Os Zb Zabbix	111 En Dd Datadog	112 Os Ei Elasticsearch	113 Fm Ad AppDynamics	114 En Sp Splunk	115 Fm Le Logentries	116 Fm Sl Sumo Logic	117 Os Ls Logstash	118 Os Sn Snort	119 Os Tr Tripwire	120 En Ff Fortify

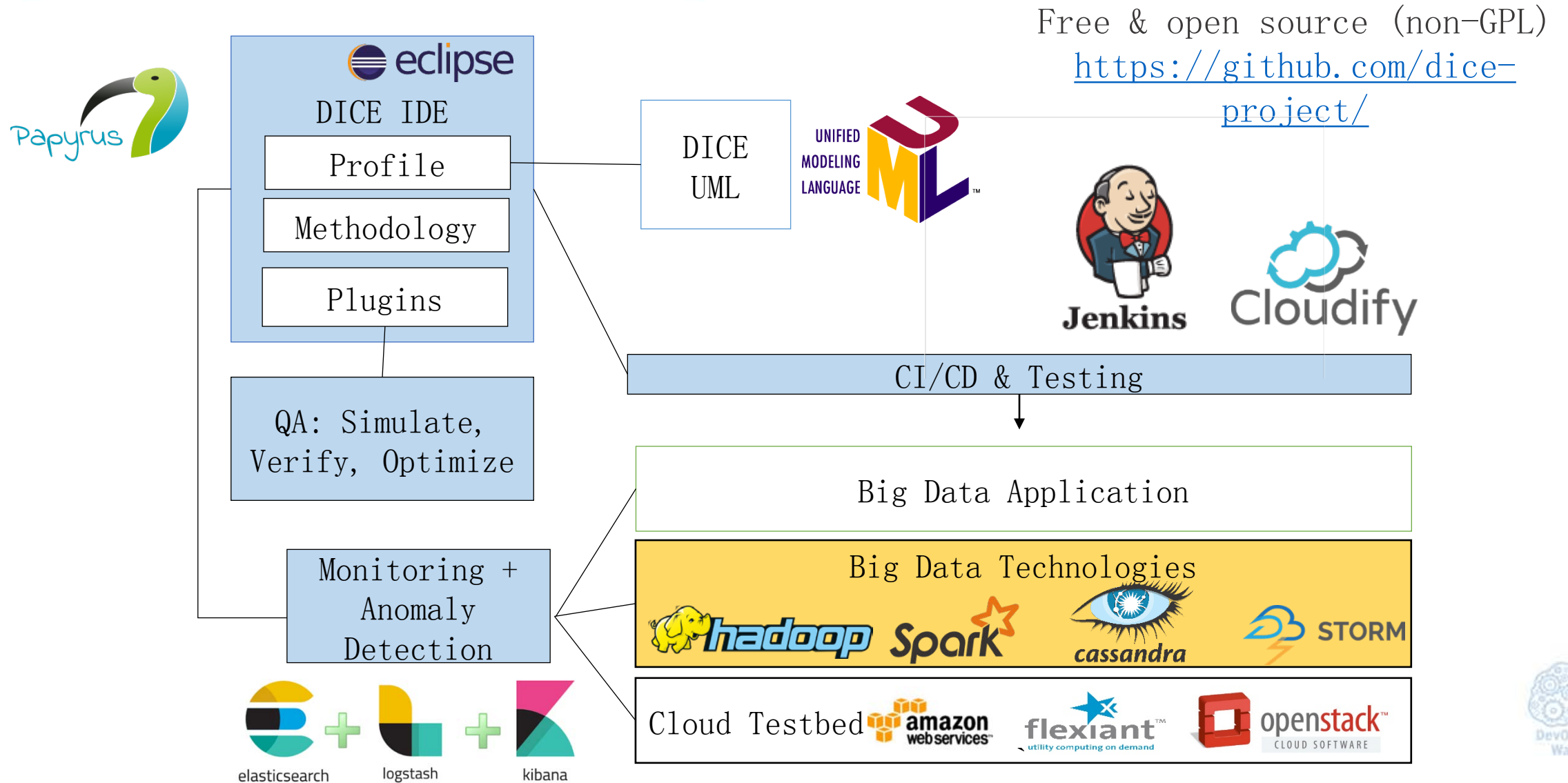
Xebialabs

Follow @xebialabs

source: xebialabs.com



The DICE Framework: Model-Driven DevOps



Dev View

workspace - DICE - prodevelop-case-study/posidonia.di - DICE Platform

File Edit Diagram Navigate Search Papyrus Project Run DICE-LWDG Deployment Service DICE Tools Window Help

Simulation Tools
DICE Monitoring Service
DICE Monitoring Service Visualization UI
DICE Optimization Service
DICE Simulation Tool
DICE Verification Tool

Project Explorer

- prodevelop-case-study
 - generated_blue
 - posidonia
 - di
 - notation
 - uml
 - yaml
 - generated_blue.tar
 - generated_blue.tar.gz
 - simu

*posidonia.di

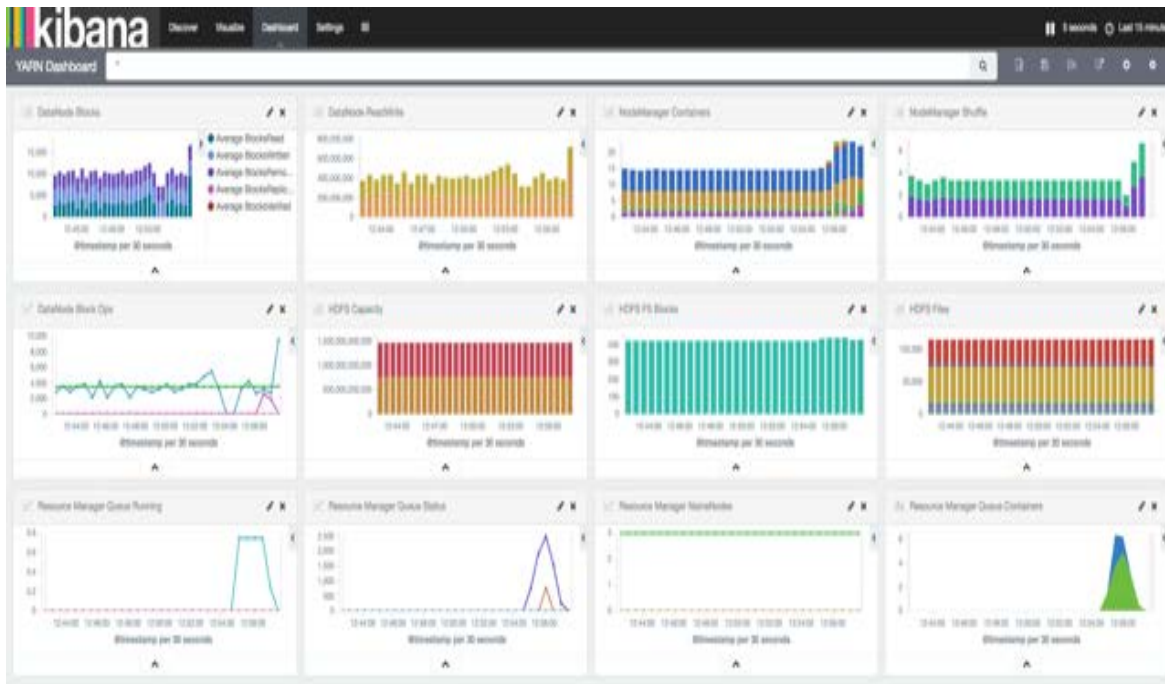
The diagram illustrates the architecture of the posidonia.di project. It consists of four main components, each represented as a VM (Virtual Machine) containing an internal component and several artifacts:

- data_parser_vm**: Contains the `data_parser` component and artifacts: `bootstrap_par...`, `prerequisites`, `install-parser`, `ipoms-ais`, `install-simu`, `ipoms-simu`, and `common`.
- rabbitmq_vm**: Contains the `rabbitmq` component and artifact: `install-rab...`.
- cep_vm**: Contains the `cep` component and artifacts: `bootstrap_c...`, `prerequisites`, and `ipoms-cep`.
- web_app_vm**: Contains the `web_app` component.

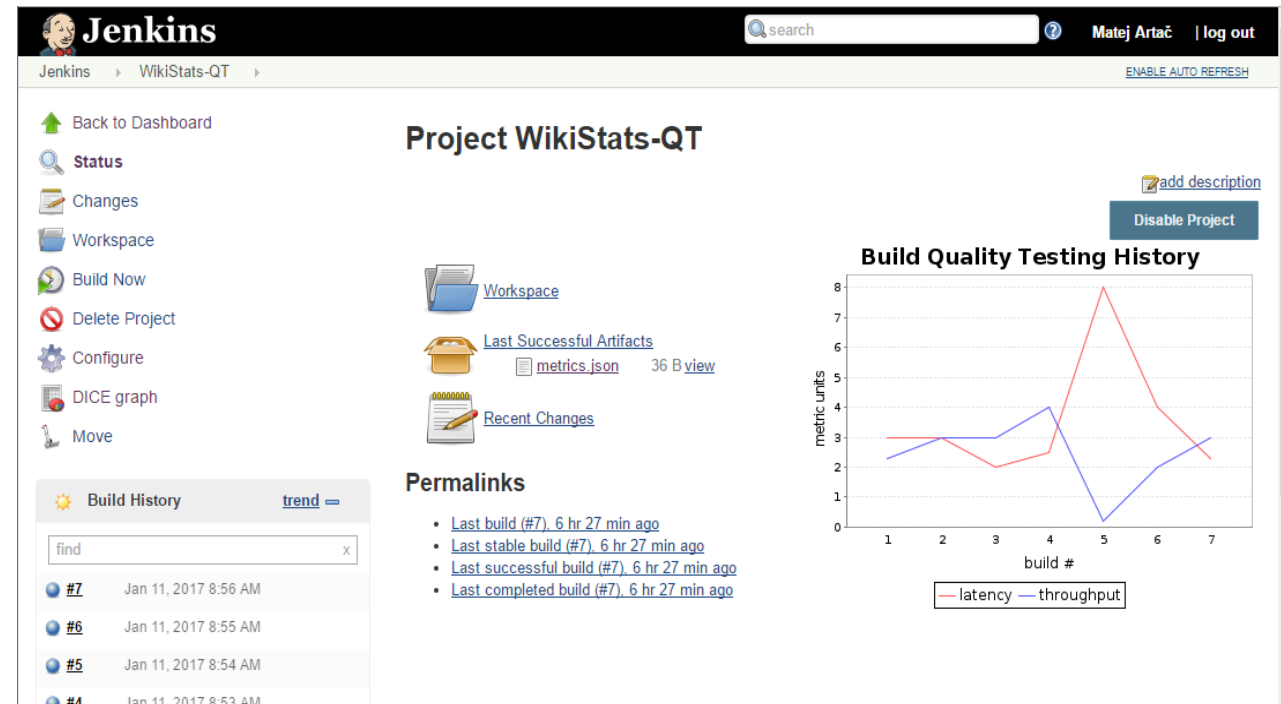
Dashed lines indicate dependencies or relationships between these components across the different VMs.

Ops Views

Kibana



Jenkins testing



Waiting step since: Wednesday, November 23, 2016 8:27 AM

Cloudify



Behind the Scenes: Dev side

DICE Platform Independent Model
(DPIM)



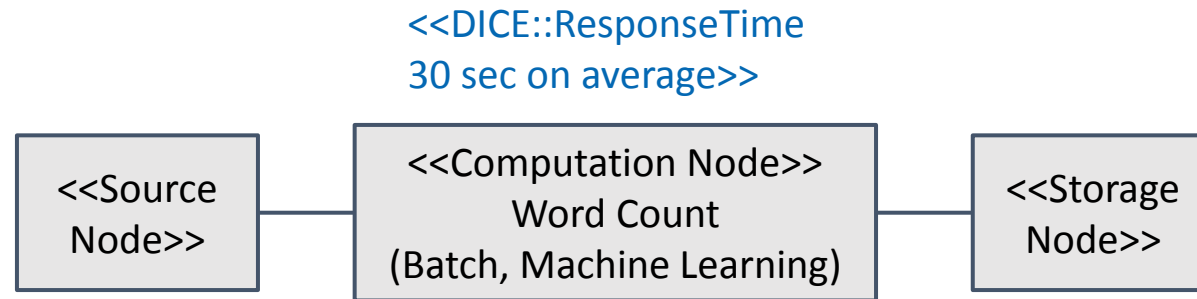
Simulate
&
Verify

DICE Methodology

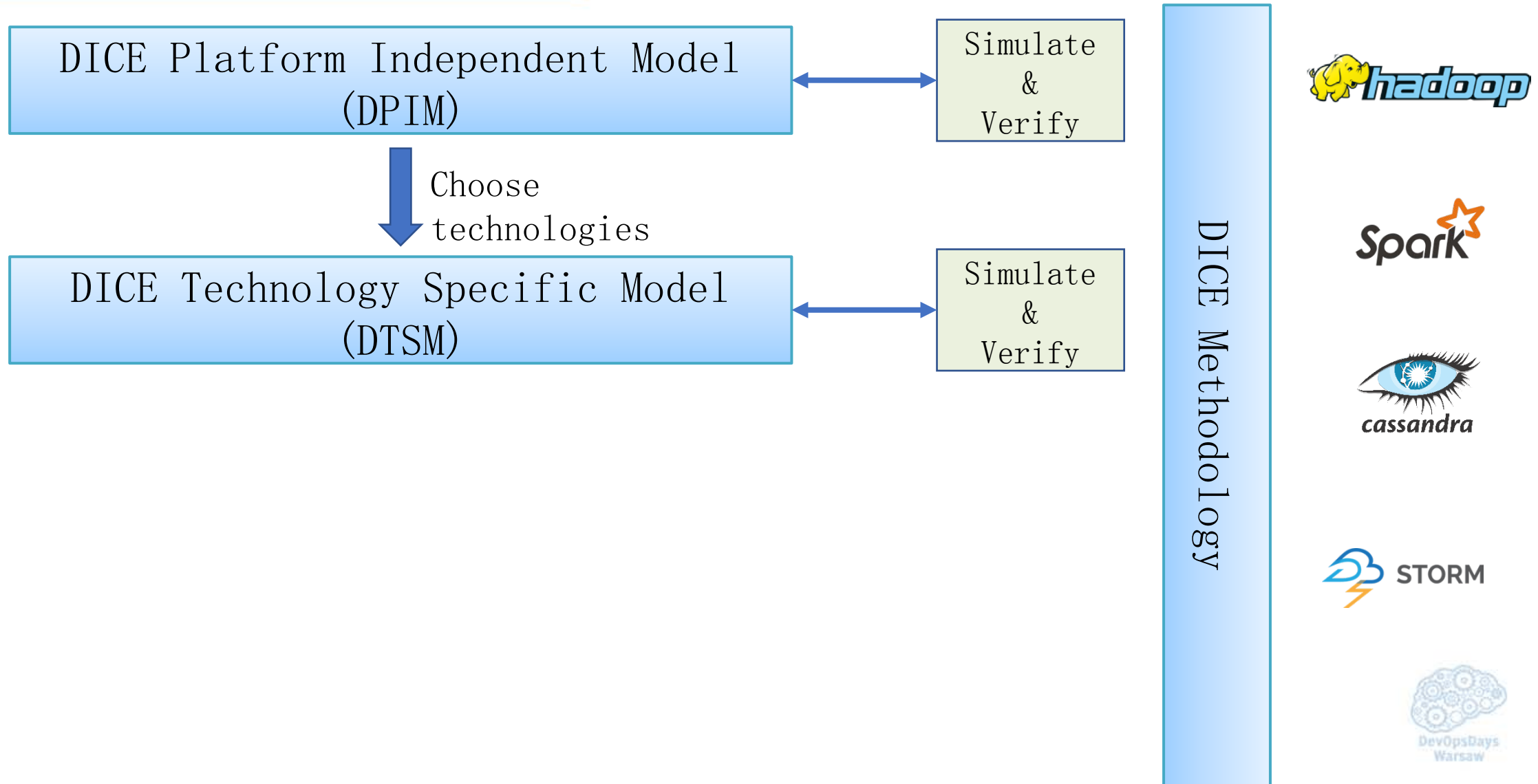


Wordcount example: platform independent model

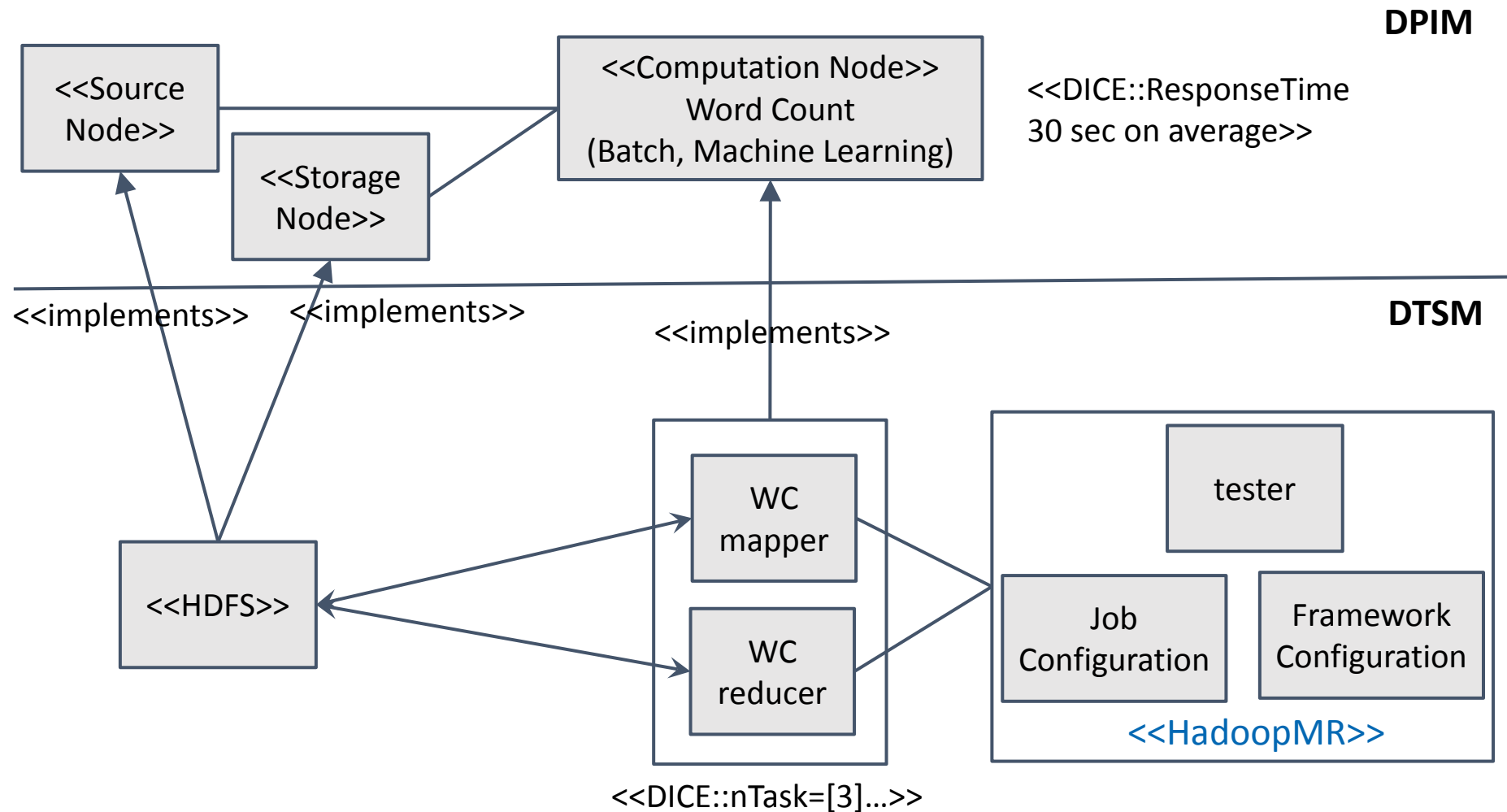
DPIM



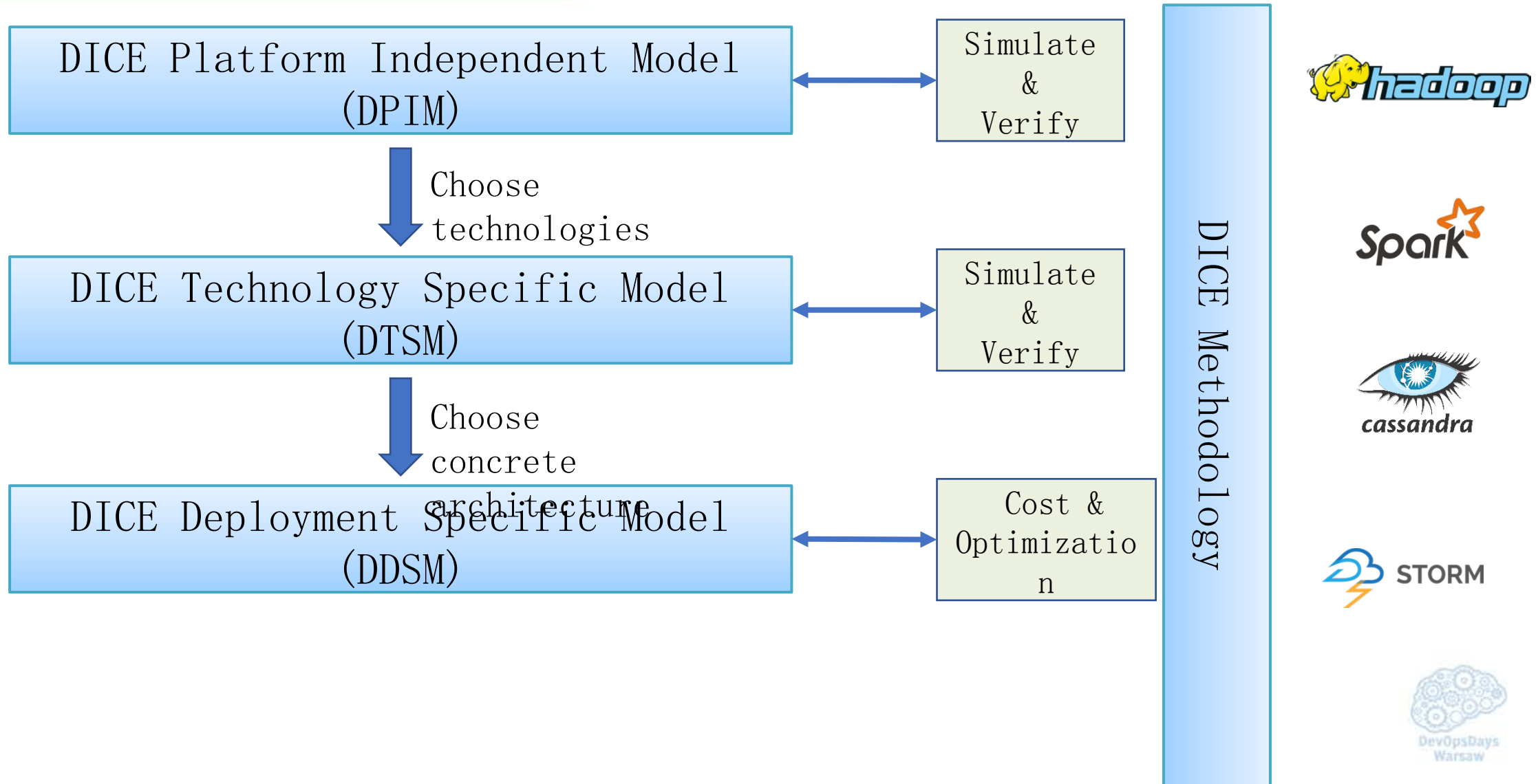
Behind the Scenes: Dev side



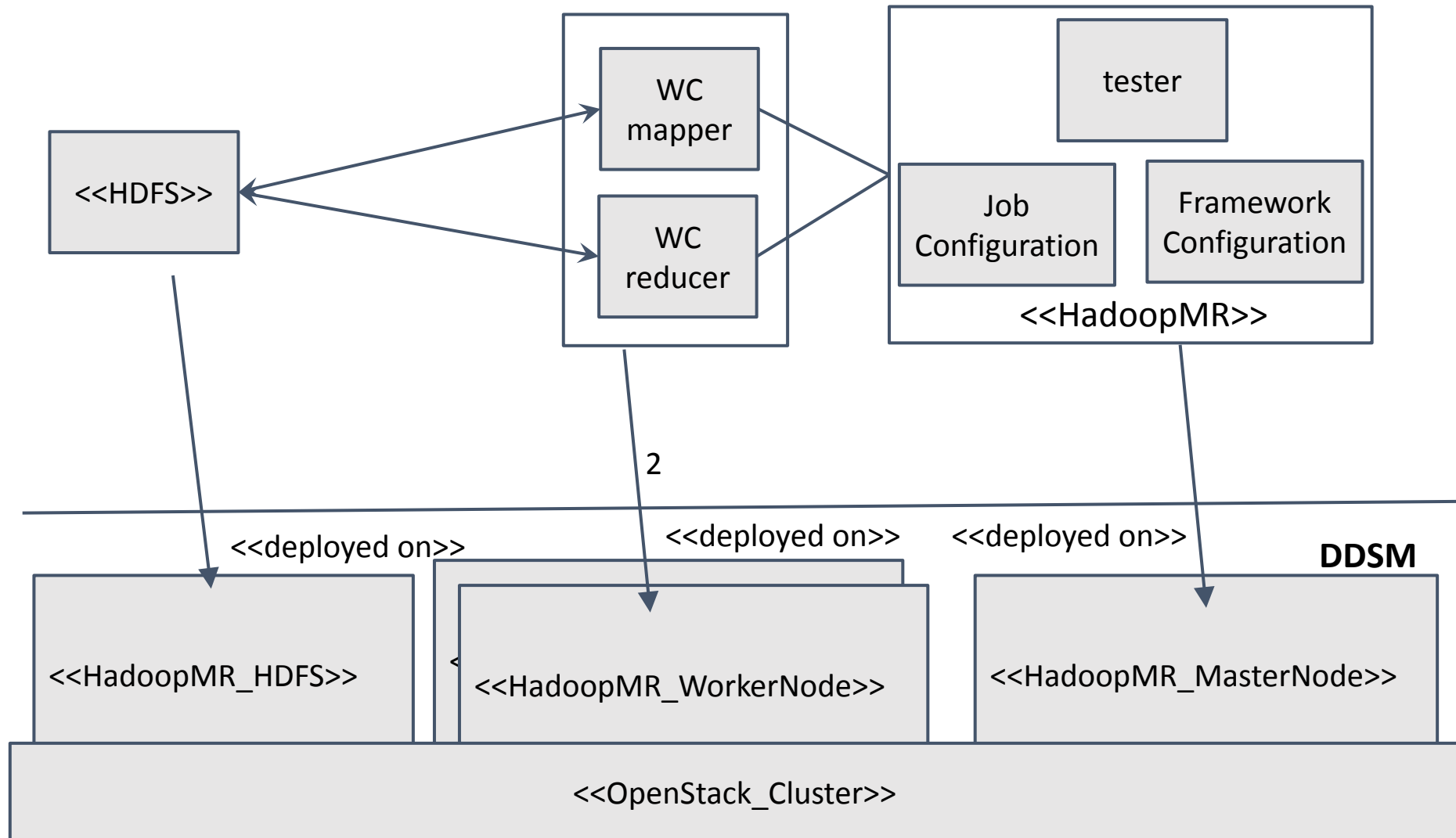
Wordcount example: technology specific model



Behind the Scenes: Dev side

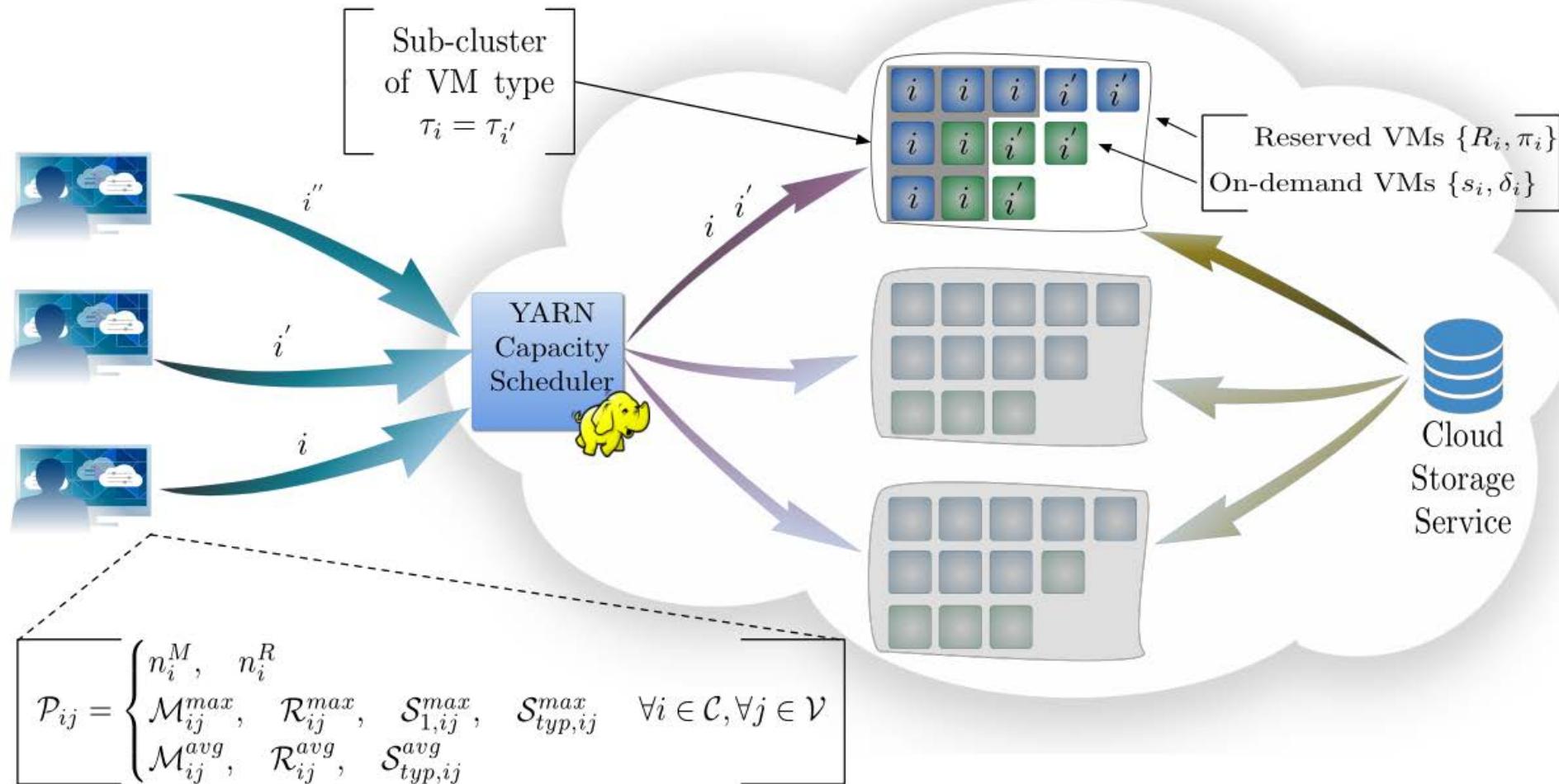


Wordcount example: deployment specific model

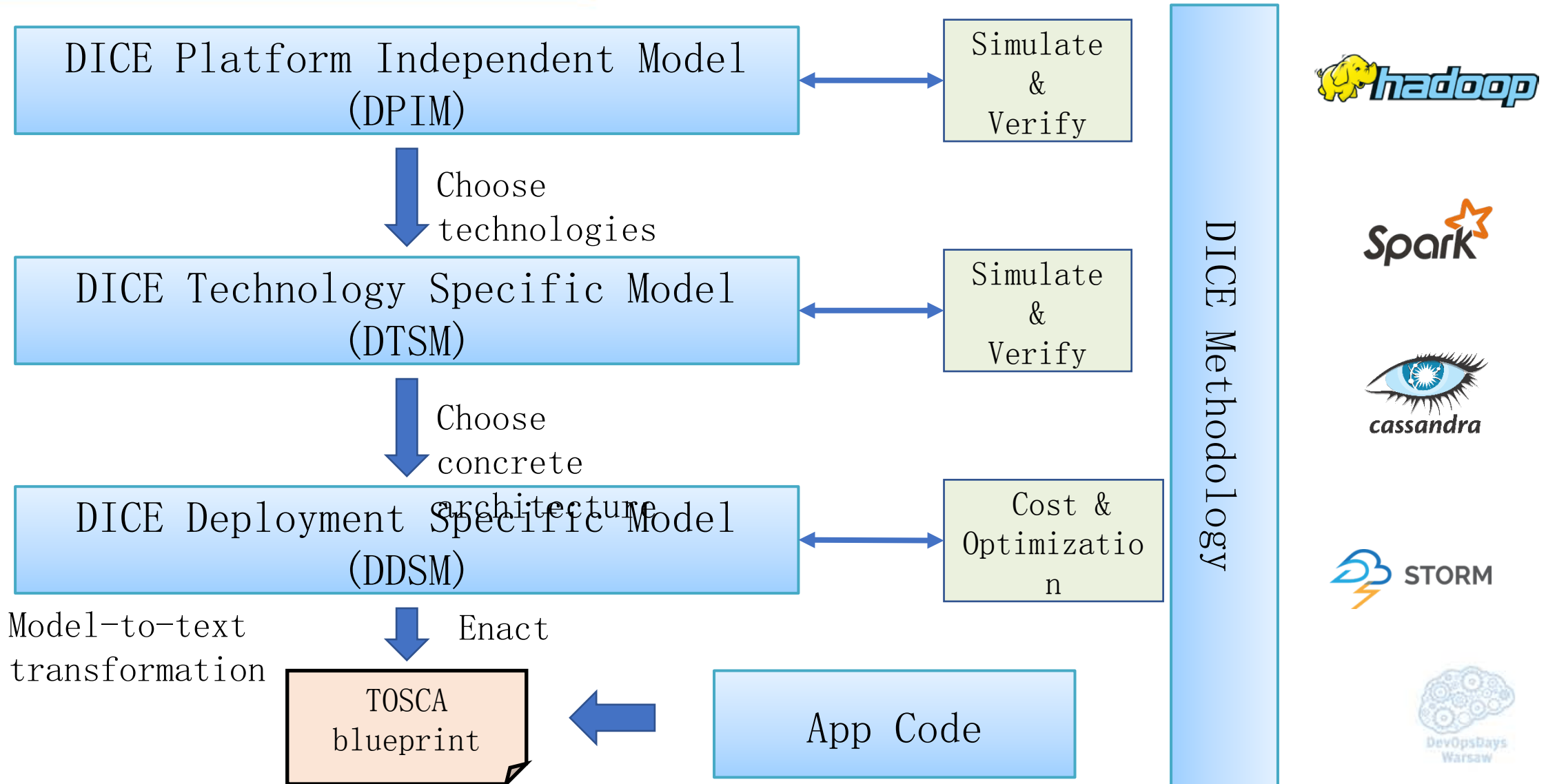


Simulation, Optimization, Verification

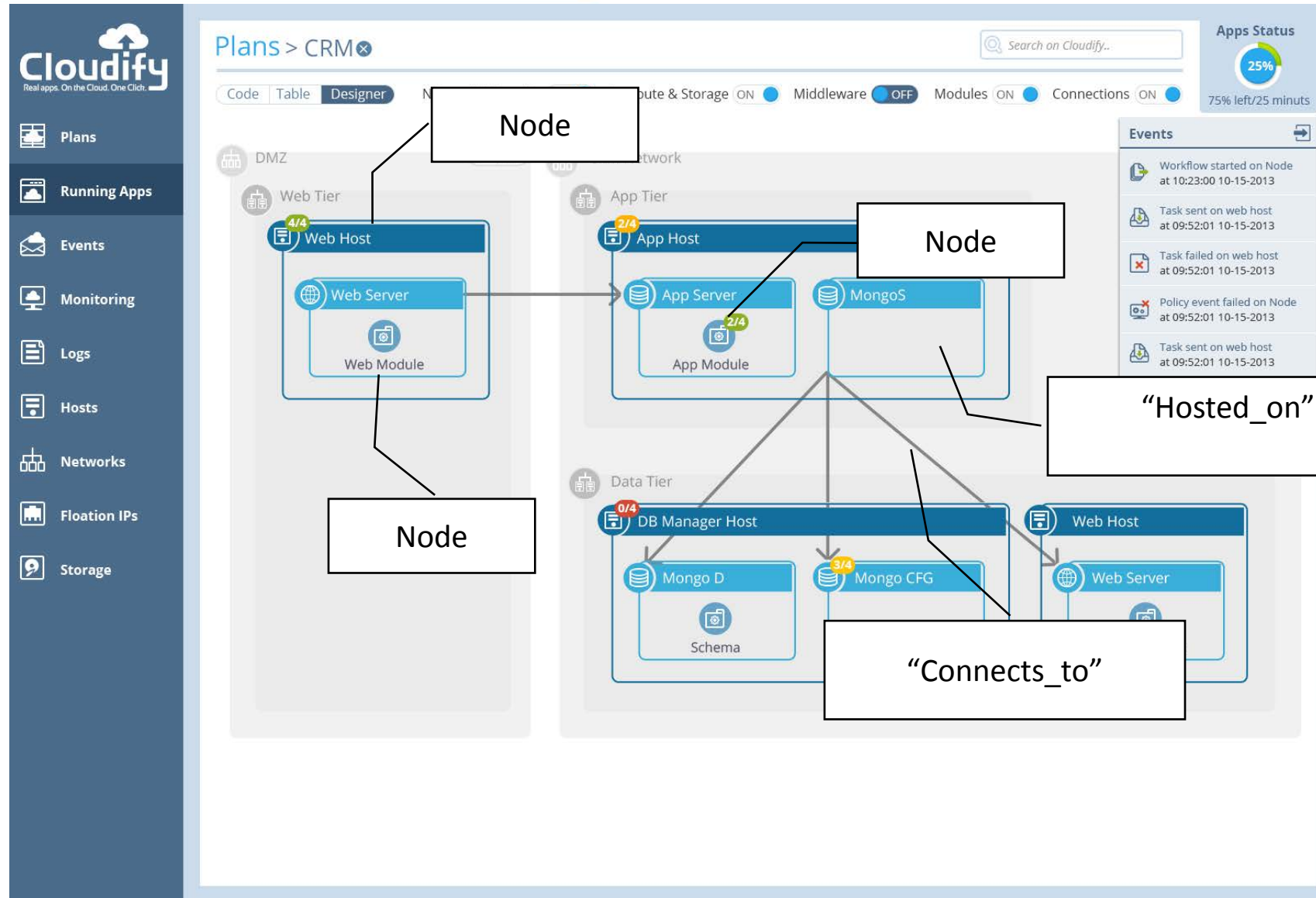
Example: Cost Optimization



Behind the Scenes: Dev side



Deployment: OASIS TOSCA & Cloudify



DICE TOSCA blueprints



Powered by
Cloudify

tosca_definitions_version: cloudify_dsl_1_3

imports: [
 'http://dice-project.github.io/DICE-Deployment-Cloudify/spec/openstack/0.1.4/plugin.yaml']

node_templates:

ZookeeperVM:

type: dice.hosts.Small
instances: {deploy: 1}
relationships:
- {type: dice.relationships.ProtectedBy, target: ZookeeperCluster_ZookeeperVM_worker_firewall}

StormMasterVM:

type: dice.hosts.Small
relationships:
- {type: dice.relationships.ProtectedBy, target: StormCluster_firewall}

StormWorkerVM:

type: dice.hosts.Small
instances: {deploy: 1}

StormCluster:

type: dice.components.storm.Nimbus

relationships:
- {type: dice.relationships.ContainedIn, target: StormMasterVM}

StormCluster:

type: dice.components.storm.Nimbus

relationships:
- {type: dice.relationships.storm.ConnectedToZookeeperQuorum, target: ZookeeperCluster}

relationships:
- {type: dice.relationships.ContainedIn,

target: StormMasterVM}

relationships:
- {type: dice.relationships.storm.ConnectedToZookeeperQuorum,

target: ZookeeperCluster}

properties:

configuration: {taskTimeout: '30', supervisorTimeout: '60',
monitorFrequency: '10', queueSize: '100000',
retryTimes: '5', retryInterval: '2000'}

StormCluster_firewall:

type: dice.firewall_rules.storm.Nimbus

StormCluster_StormWorkerVM_slave:

type: dice.components.storm.Worker

relationships:
- {type: dice.relationships.ContainedIn, target: StormWorkerVM}

relationships:
- {type: dice.relationships.storm.ConnectedToZookeeperQuorum, target: ZookeeperCluster}

properties:

configuration: {workerTaskTimeout: '30', heartbeatFrequency: '5', cpuCapacity: '400',
memoryCapacity: '4096'}

ZookeeperCluster_ZookeeperVM_worker_firewall:

type: dice.components.zookeeper.Server

relationships:
- {type: dice.relationships.ContainedIn, target: ZookeeperVM}

relationships:
- {type: dice.relationships.zookeeper.MemberOfQuorum, target: ZookeeperCluster}

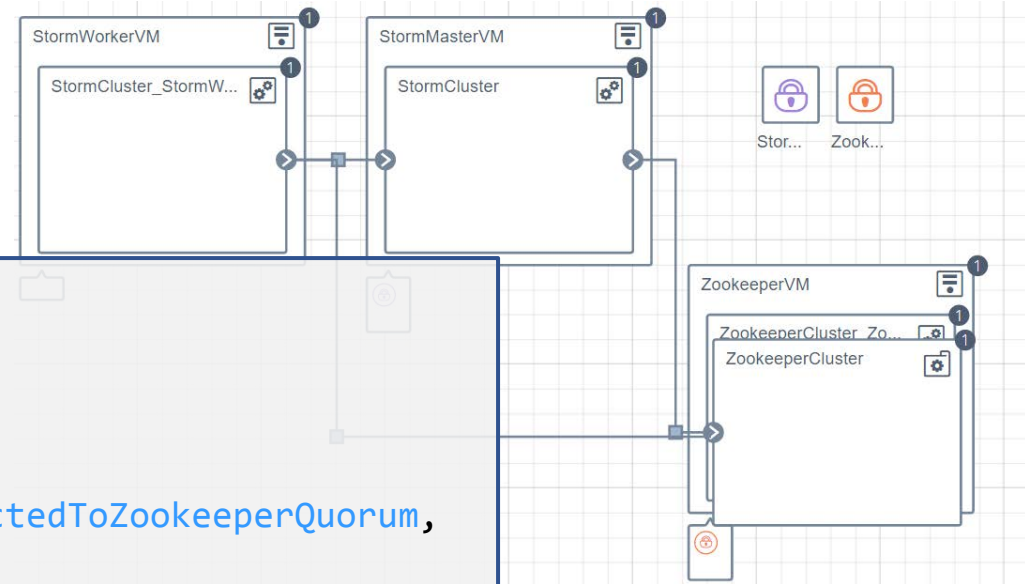
properties:

configuration: {tickTime: '1500', initLimit: '10', syncLimit: '5'}

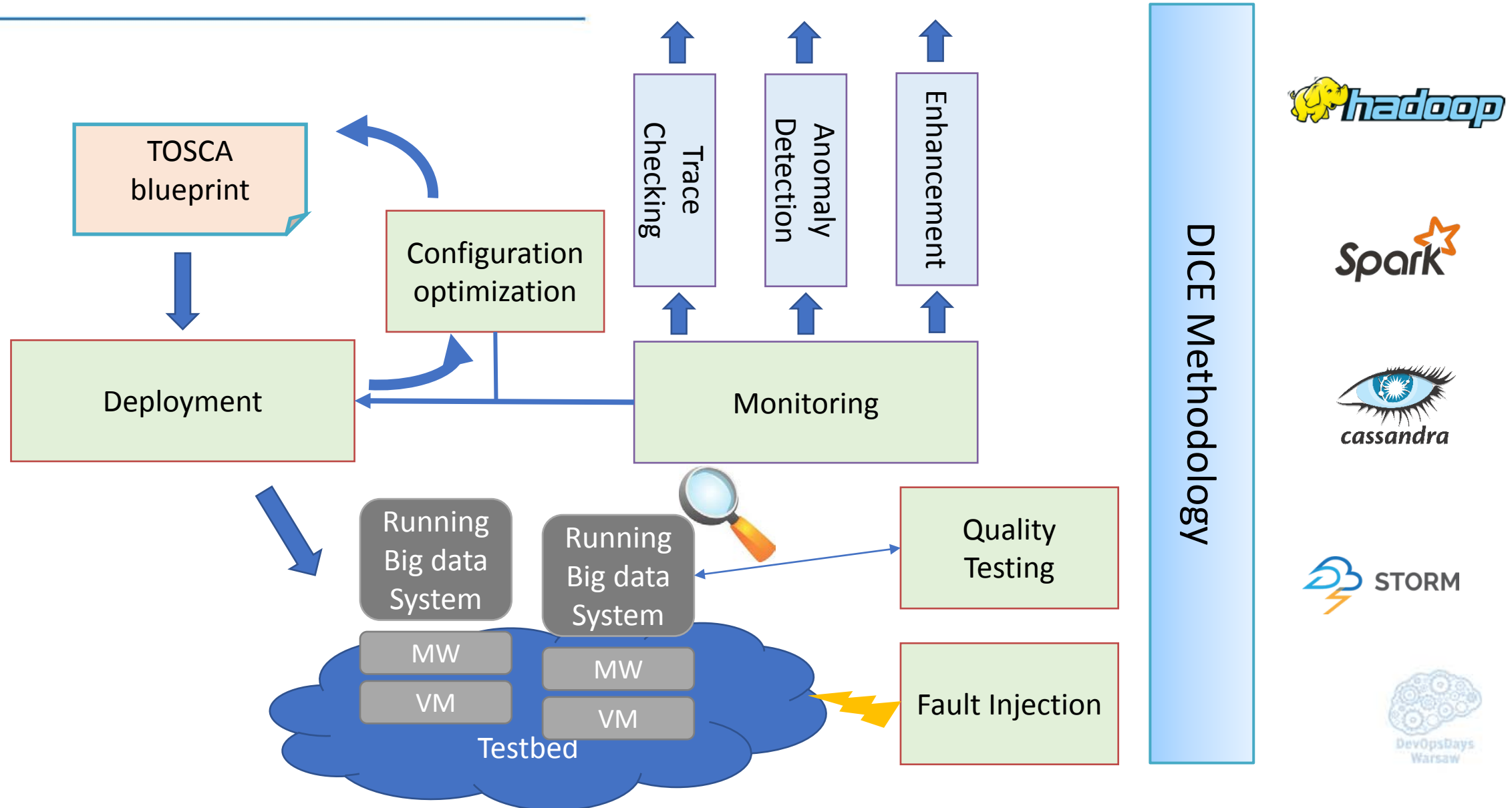
ZookeeperCluster:

type: dice.components.zookeeper.Quorum

relationships:
- {type: dice.relationships.zookeeper.QuorumContains, target: ZookeeperVM}



Behind the Scenes: Ops side



DICE Ops Tools: configuration optimization for Big data



Example: 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
```

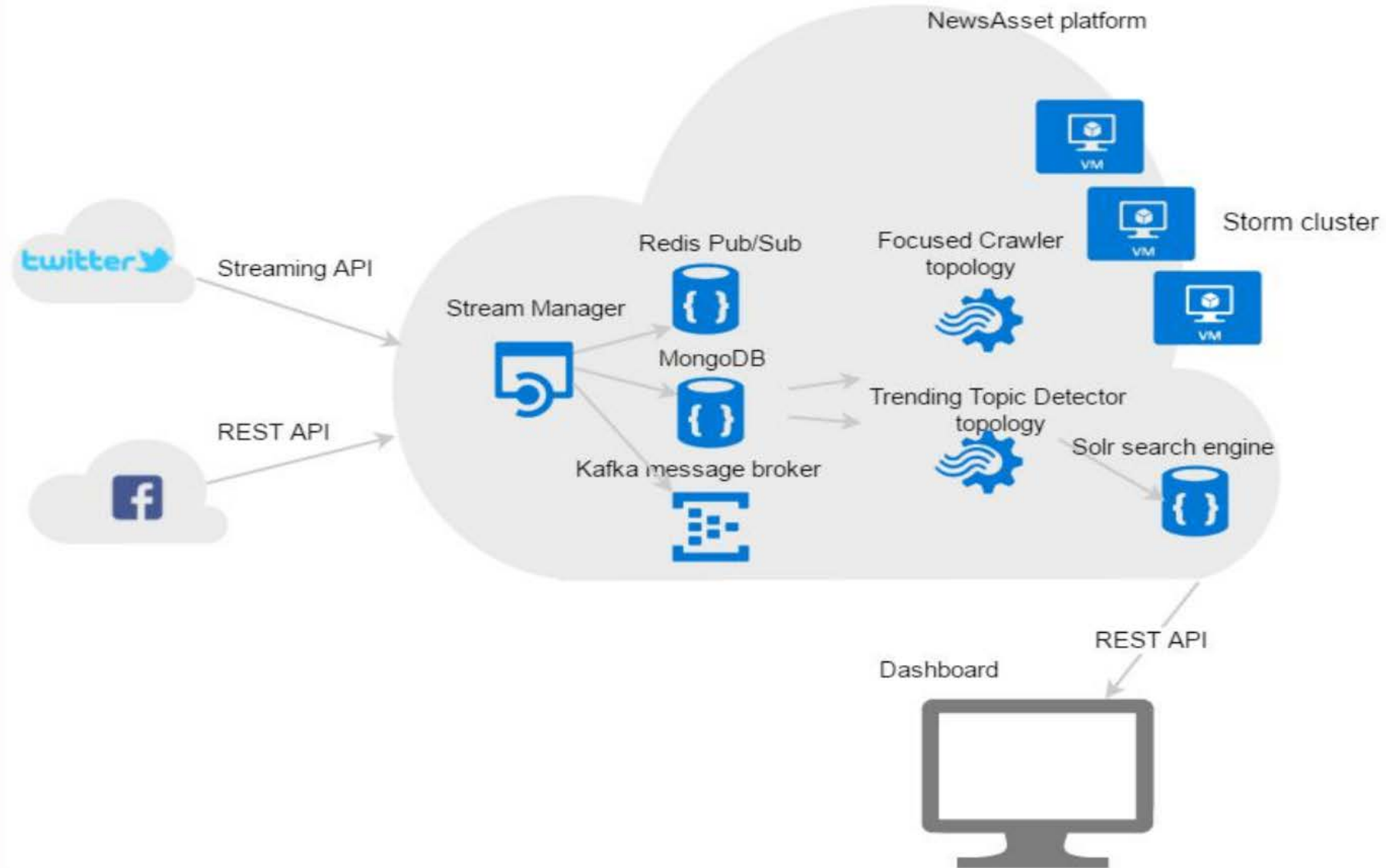


NewsAsset platform

Content Creation & Editorial workflow



News & social media analysis



DICE configuration optimization

The image displays the DICE Configuration View interface, which includes a topology diagram on the left and a configuration table on the right.

Topology Diagram:

- Partition1:** Contains a Storm Spout (`spout_1`) and a Storm Bolt (`bolt_1`). The spout has a configuration of `{parallelism=$n0, hostDemand=(expr=$sp_1, unit=ms, statQ=mean, source=est)}`. It connects to a Storm Bolt via a Storm Stream Step with `{numTuples=$nS6, grouping=all}`. The bolt has a configuration of `{parallelism=$n4; hostDemand=(expr=$b_1, unit=ms, statQ=mean, source=est)}`.
- Partition2:** Contains a Storm Spout (`spout_2`) and a Storm Bolt (`bolt_2`). The spout has a configuration of `{parallelism=$n1, hostDemand=(expr=$sp_2, unit=ms, statQ=mean, source=est)}`. It connects to a Storm Bolt via a Storm Stream Step with `{numTuples=$nS2, grouping=shuffle}`. The bolt has a configuration of `{parallelism=$n2; hostDemand=(expr=$b_2, unit=ms, statQ=mean, source=est)}`.
- External Connections:** A top-level configuration `{utilization=(expr=$use, unit=%, statQ=mean, source=calc)}` is connected to `spout_1`. `spout_1` also connects to `BM1` (a Bolt Merge) in Partition2 via a Storm Stream Step with `{numTuples=$nS1, grouping=shuffle}`. `BM1` connects to `bolt_2` via a Storm Stream Step with `{numTuples=$nS5, grouping=all}`.

DICE Configuration View Interface:

The interface includes tabs for `Properties`, `Console`, `SVN Repositories`, `Git Repositories`, `Invocations Registry`, and `DICE Configuration View`. The `Plugin Config` tab is active, showing a dropdown menu for `storm`.

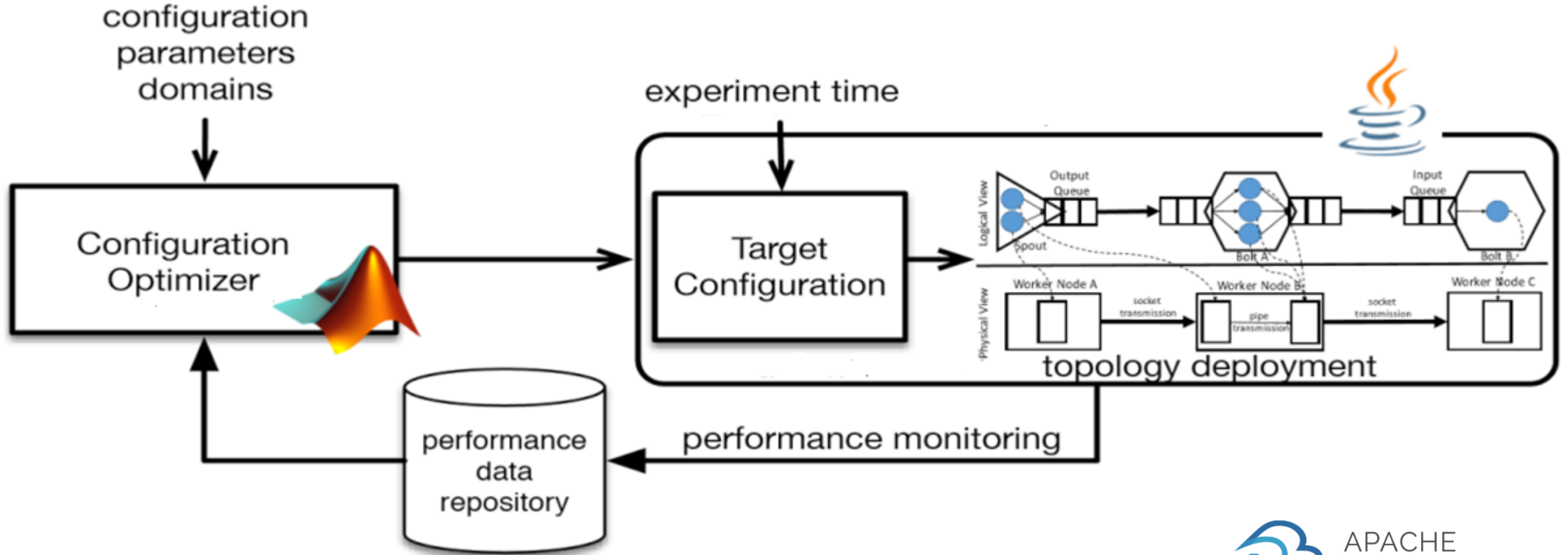
Parameter Selection Table:

Parameter	Description
<code>topology.error.throttle.interval.secs</code>	
<code>topology.trident.batch.emit.interval.millis</code>	
<code>topology.disruptor.wait.timeout.millis</code>	
<code>topology.disruptor.batch.size</code>	
<code>topology.disruptor.batch.timeout.millis</code>	
<code>topology.disable.loadaware.messaging</code>	
<code>topology.state.checkpoint.interval.ms</code>	
<code>topology.max.spout.pending</code>	
<code>topology.acker.executors</code>	
<code>topology.tick.tuple.freq.secs</code>	

Add Parameters Table:

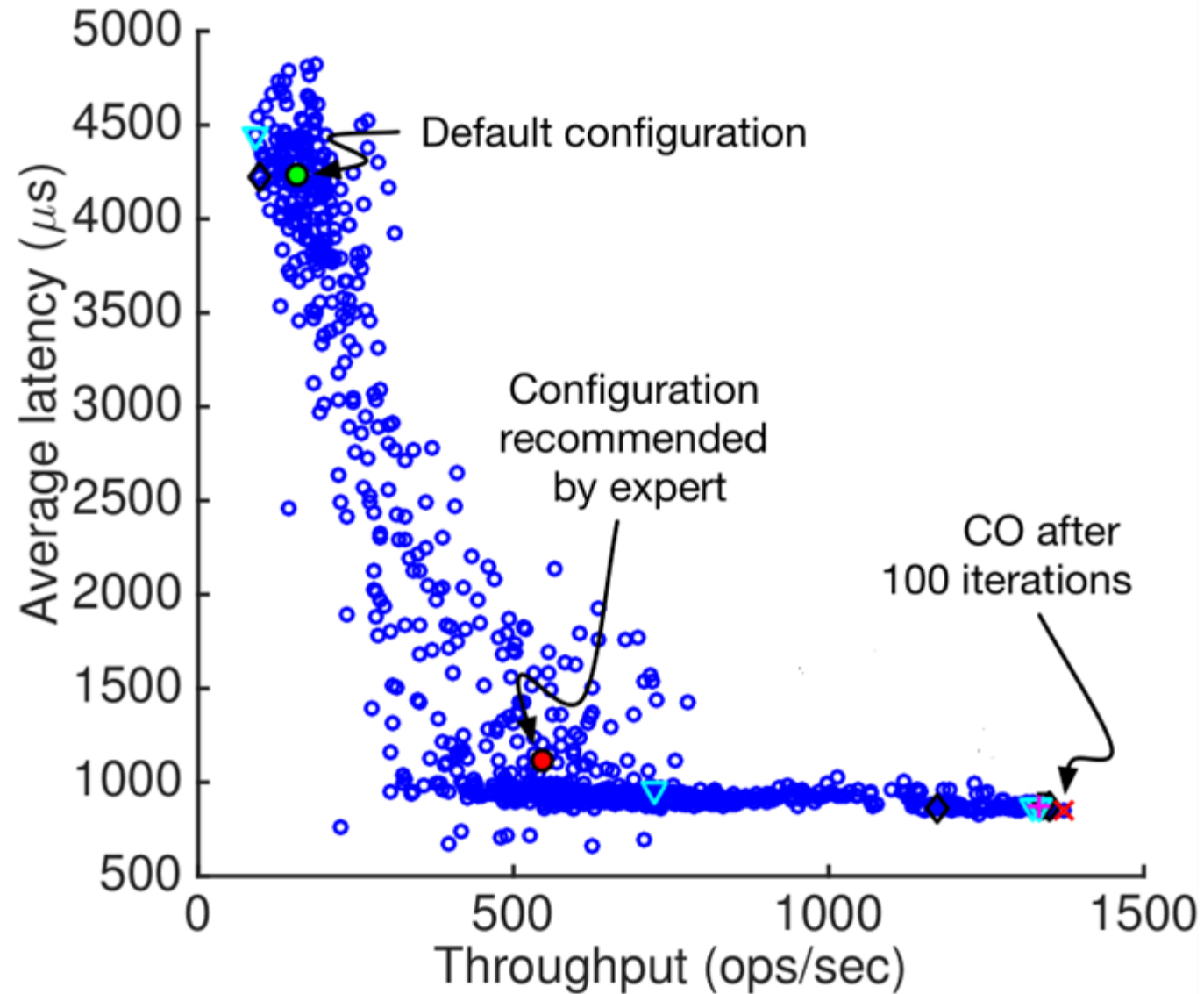
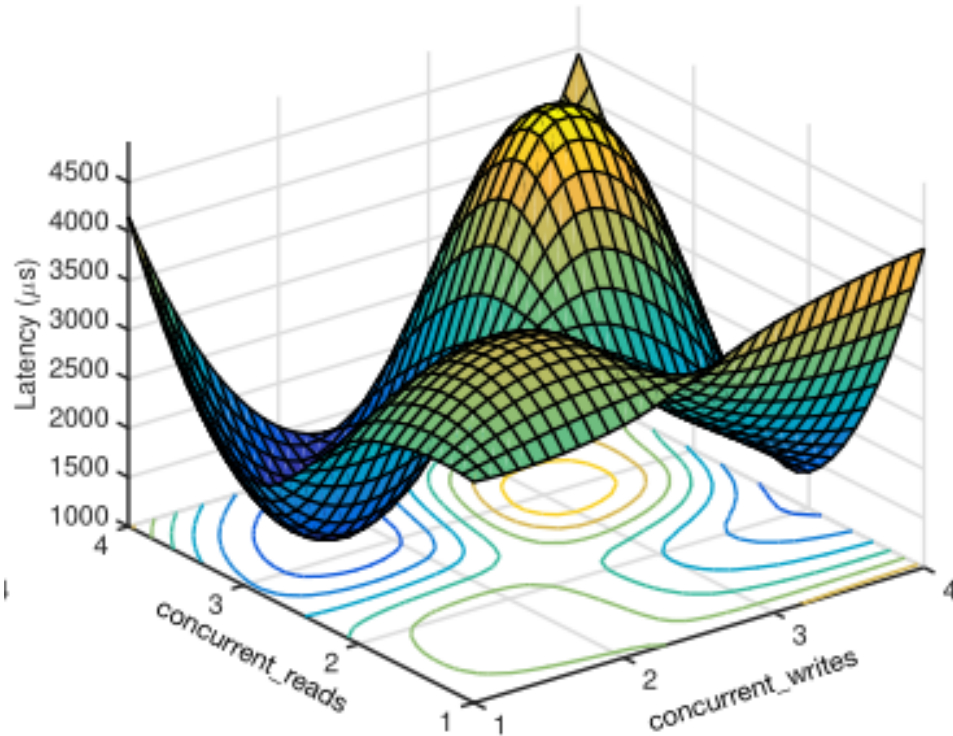
Parameter	Type	Min	Max	Step	Options
<code>topology.executor.receive.buffer.size</code>	Integer	1024	2048	1	
<code>topology.min.replication.count</code>	Integer	1	10	1	
<code>topology.worker.shared.thread.pool.size</code>	Integer	1	20	1	
<code>topology.max.task.parallelism</code>	Integer	1	50	1	

DICE configuration optimization



DICE configuration optimization

Applying Bayesian Optimization to DevOps

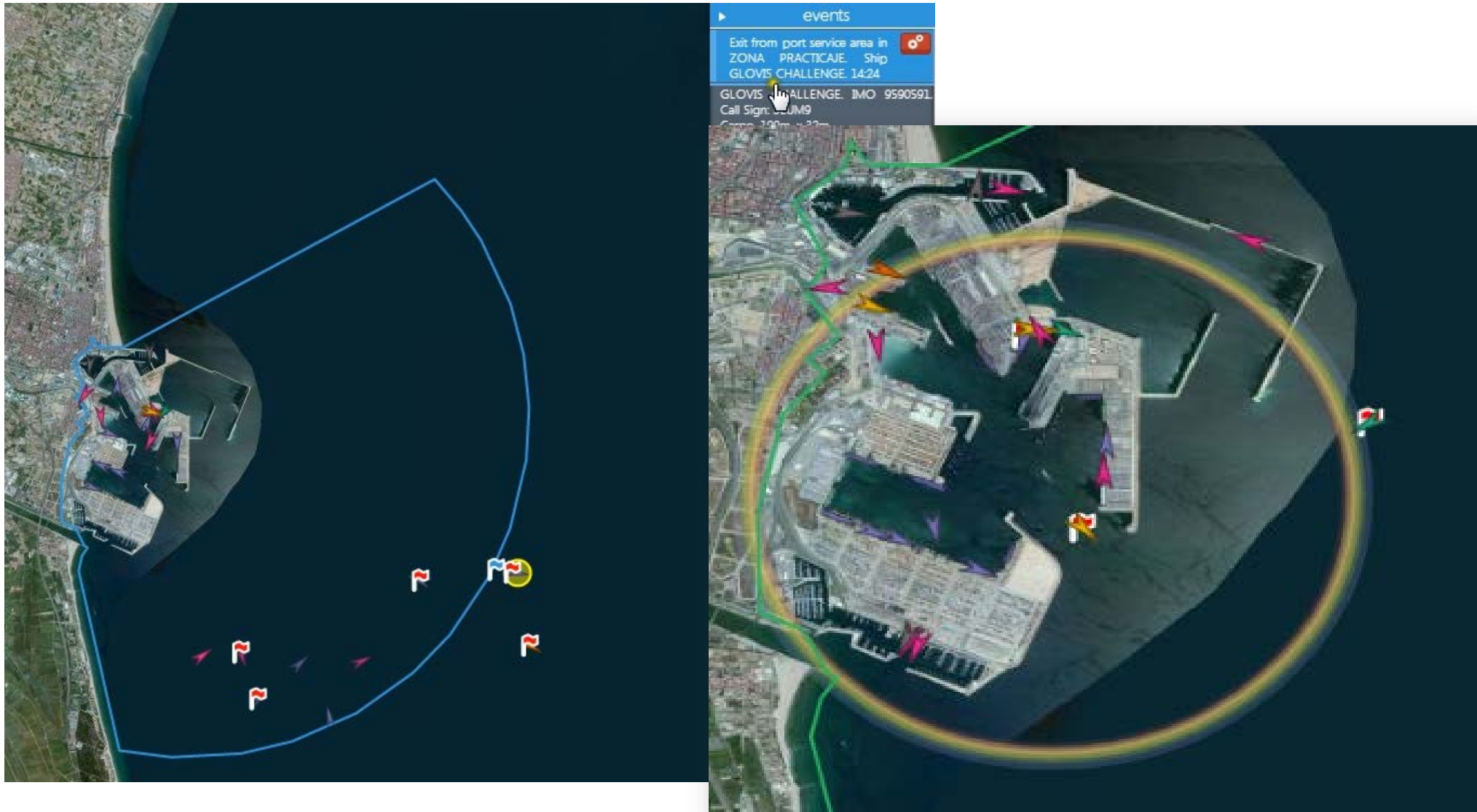


DICE Ops Tools: deployment

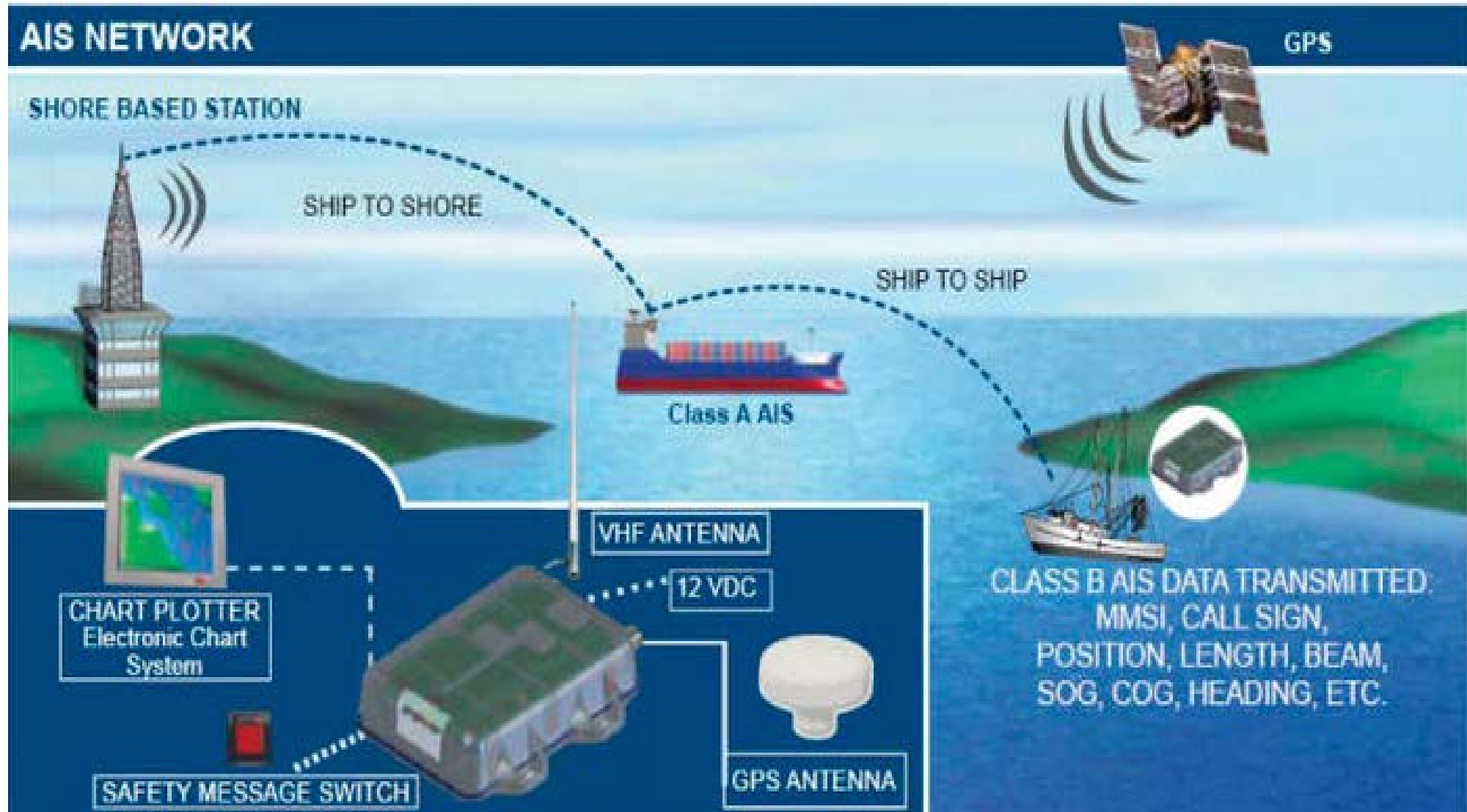


DevOpsDays
Warsaw

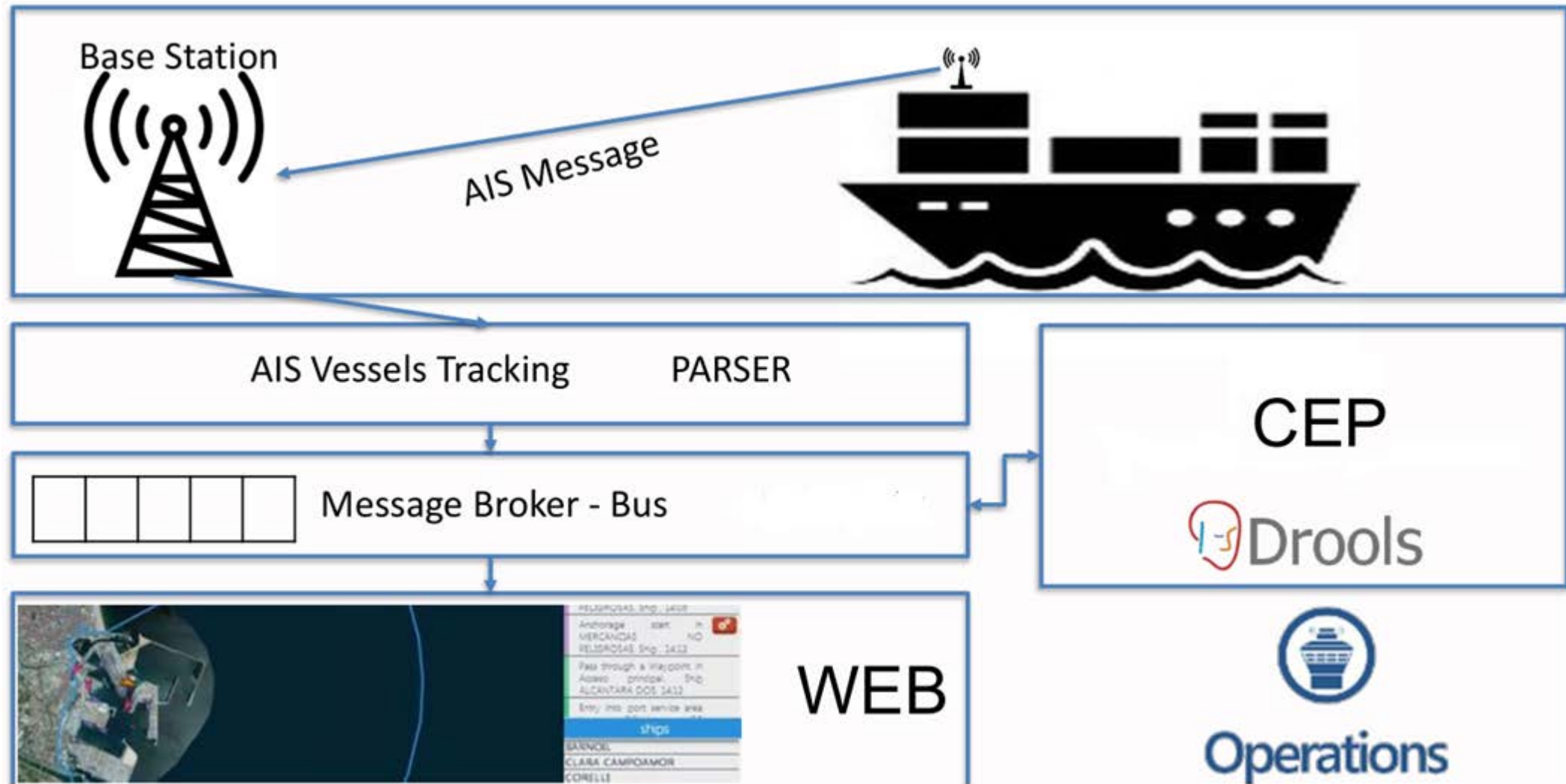
Guiding ships in ports with geofencing



The data



Posidonia Operations





In numbers

Production testbed deployment

8h

Before

5h

1st attempt

20min

Later
attempts

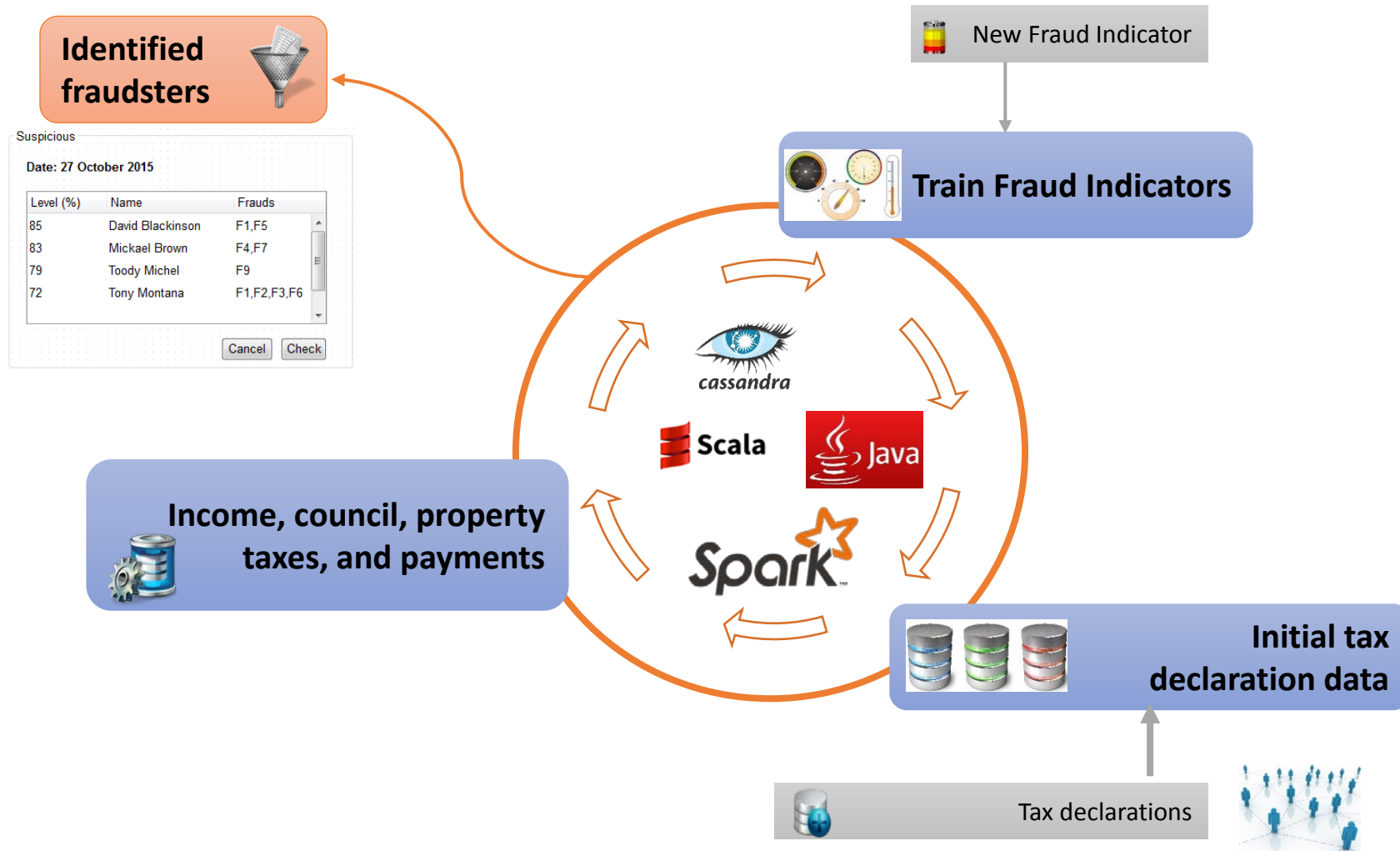
Cross-cutting concerns: data privacy



Application Domain: Tax Fraud Detection



Using Big data for Tax Fraud detection

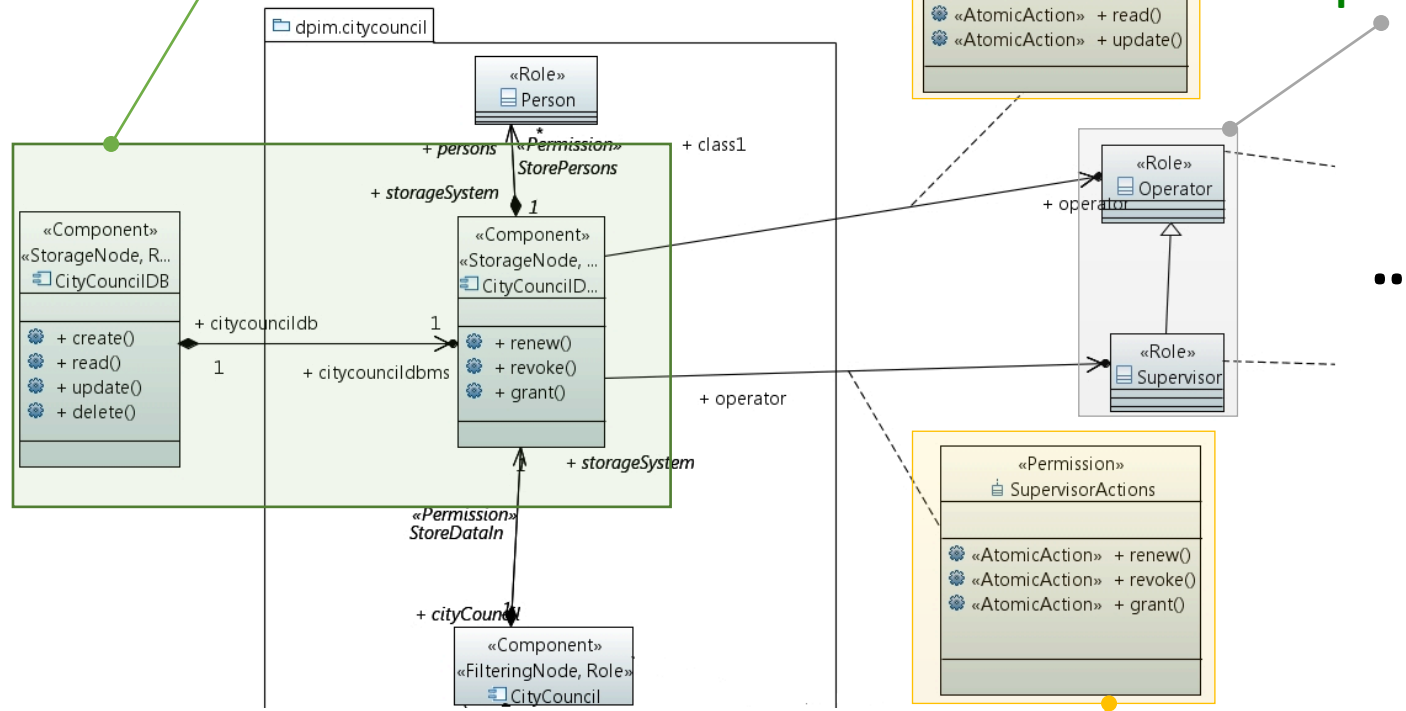


Correctness of data privacy rules

(Create, Read, Update) Permissions

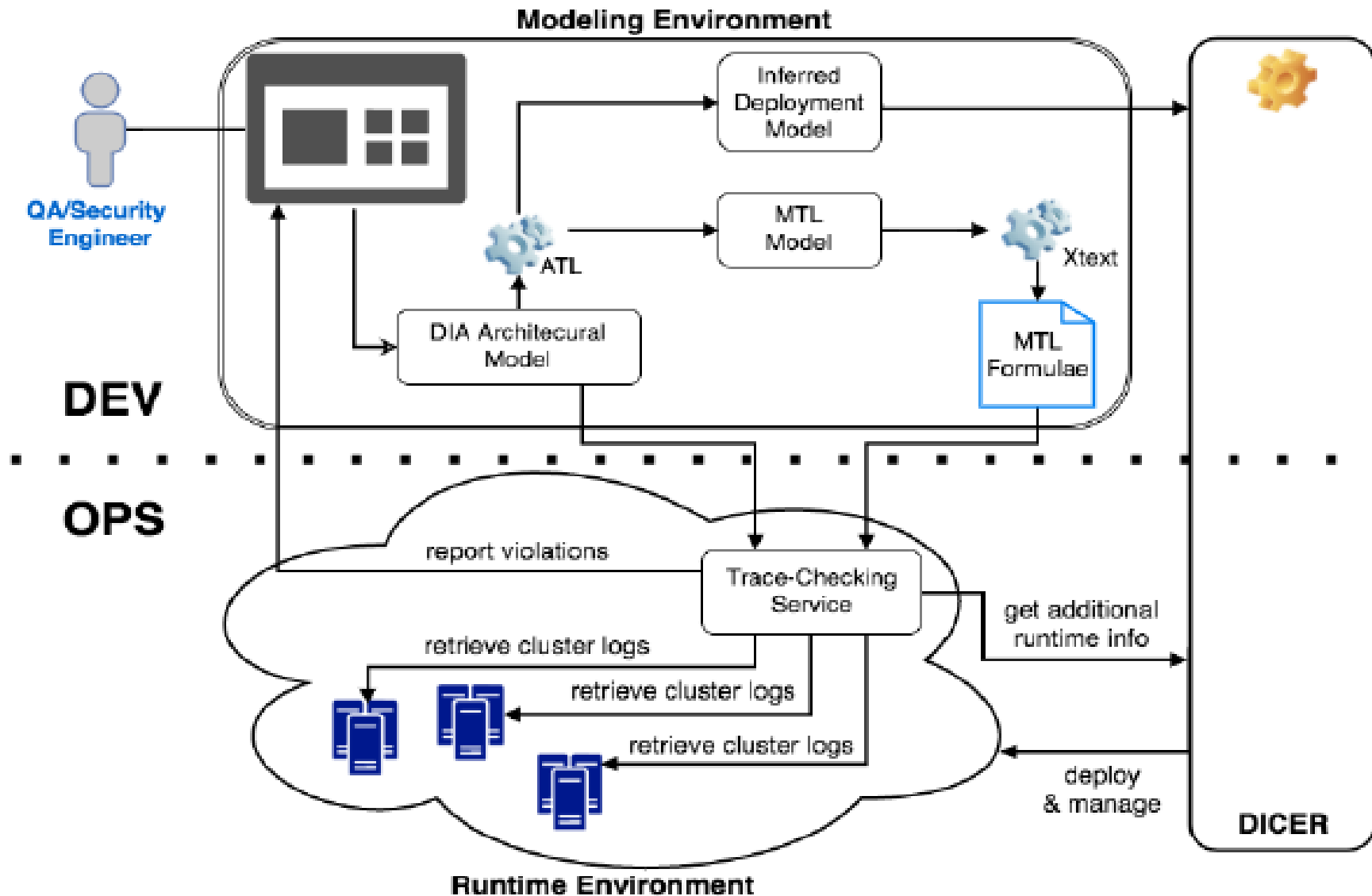
Resources (StorageNodes,...)

Roles (Operator & Supervisor)



(Grant, Renew, Revoke) Permissions

Runtime verification of data privacy rules



Conclusion

- Model-driven helps “gluing” Dev, Ops, & QA
 - Shared view of the system
 - Facilitates reasoning on and enforcing quality concerns such as privacy, performance, ...
- Automated model-driven deployment:
 - Valuable to SMEs new to DevOps and/or Big data



WWW.DICE-H2020.EU

DEVOPS FOR BIG DATA

The screenshot shows the DICE project website. The header includes the DICE logo and the tagline "Developing Data-Intensive Cloud Applications with Iterative Quality Enhancements". The navigation menu includes "Home", "About Us", "Roadmap", "Capabilities", "Partners", "Deliverables", "Contact", "Blog", and "Tools". The "Tools" section is highlighted, and the page content includes a list of tools and a diagram illustrating the DICE framework.

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 to facilitate coding, design and application prototyping, based on the Eclipse IDE, and the DICE Methodology.
- Quality analysis tools:** a set of tools for quality analysis during the early-stage of application design via simulation, verification and optimisation feedback.
- Feedback and Iterative Enhancement tools:** a monitoring platform, tailored to Big Data technologies and coupled with tools for data access and data quality analysis that allow for a global design and repair for the data-intensive application under resource.
- Continuous Delivery and Testing tools:** a set of tools and methods supporting delivery of private and public clouds via a TOSCA-compliant deployment tool, external application configuration, continuous integration, and quality testing.

The diagram below illustrates the DICE framework components and their interactions:

- DICE IDE:** Includes Eclipse DICE IDE, Profile, Methodology, and Plugins.
- Quality Analysis:** A central component that interacts with the DICE IDE and the Feedback & Iterative component.
- UMI-Based MDK:** A component that interacts with the DICE IDE and the Quality Analysis component.
- Continuous Delivery & Testing:** A central component that interacts with the Quality Analysis, Data Intensive Application (DIA), and Feedback & Iterative components.
- Data Intensive Application (DIA):** A component that interacts with the Continuous Delivery & Testing component.
- Feedback & Iterative:** A component that interacts with the Quality Analysis, Continuous Delivery & Testing, and Big Data Technologies components.
- Big Data Technologies:** A component that interacts with the Feedback & Iterative component.

The diagram also includes logos for partners and demonstrators: **Partners:** UMI-Based MDK, **Demonstrators:** Nektive, ATC, and proDEVELOP.

Q&A

GIULIANO CASALE, IMPERIAL COLLEGE LONDON

G.CASALE@IMPERIAL.AC.UK



DevOpsDays
Warsaw