A CONTAINER-CENTRIC METHODOLOGY FOR BENCHMARKING WORKFLOW MANAGEMENT SYSTEMS

CLOSER 2016

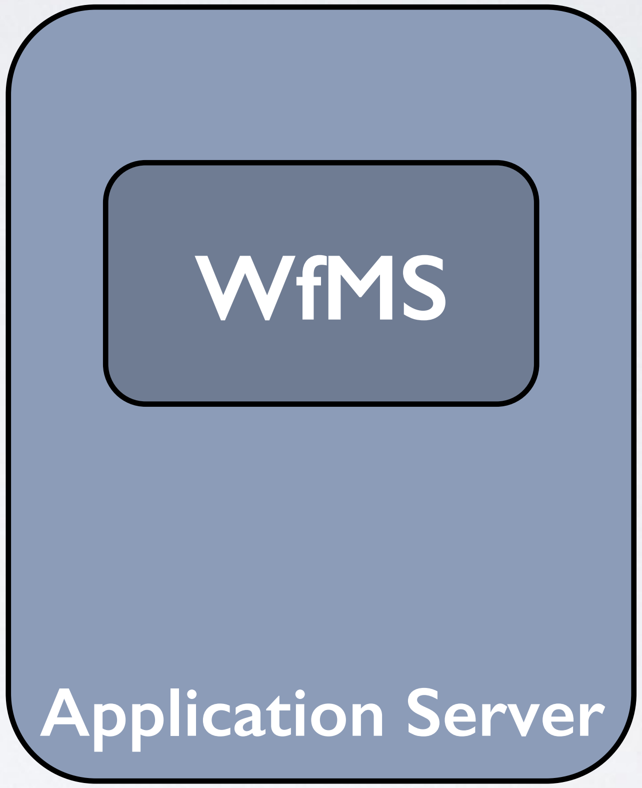
Vincenzo Ferme, Ana Ivanchikj,
Prof. Cesare Pautasso
Faculty of Informatics
University of Lugano (USI), Switzerland

Marigianna Skouradaki,
Prof. Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart, Germany

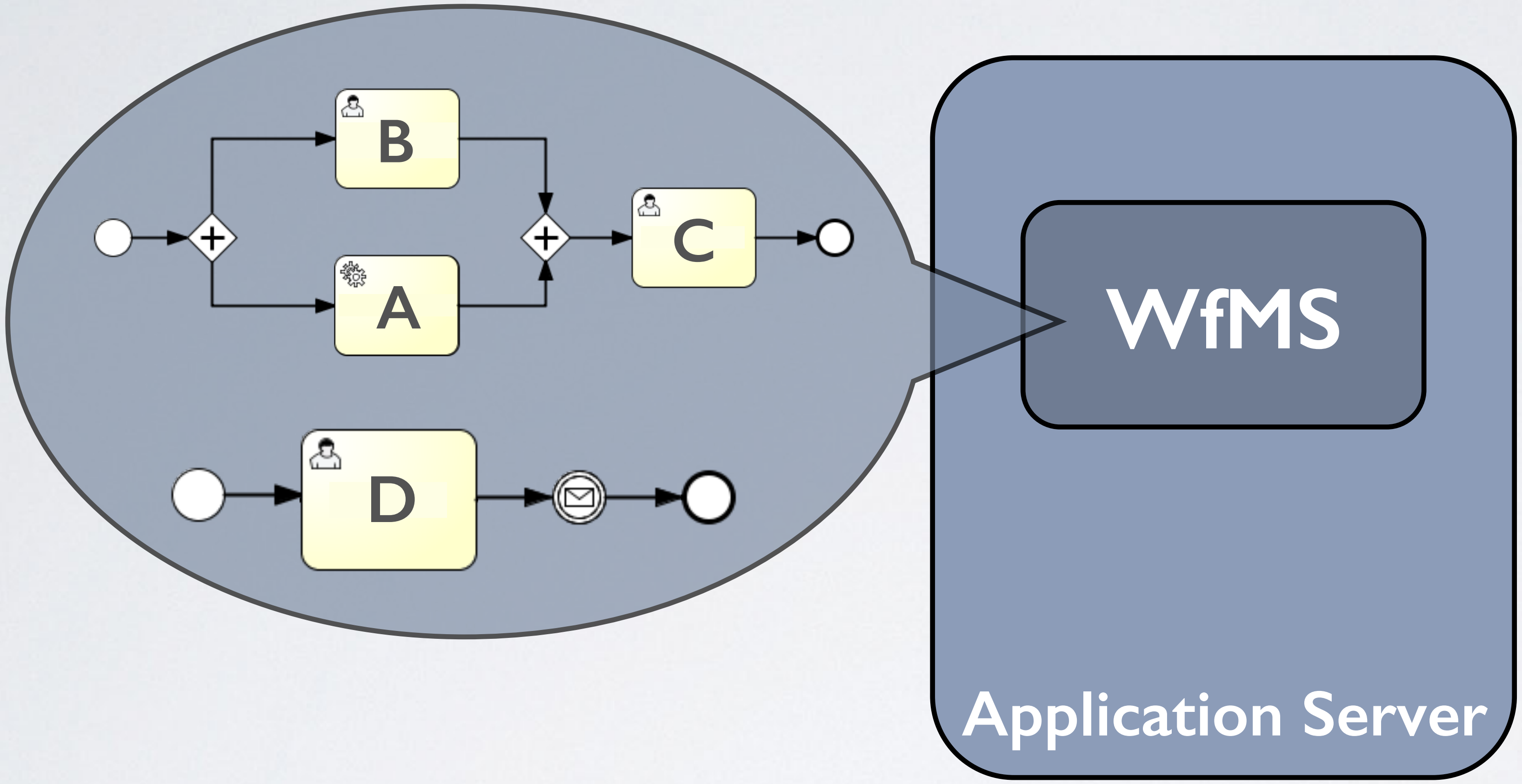
The BenchFlow Project

“Design and implement the first benchmark to assess and compare the performance of WfMSs that are compliant with Business Process Model and Notation 2.0 standard.”

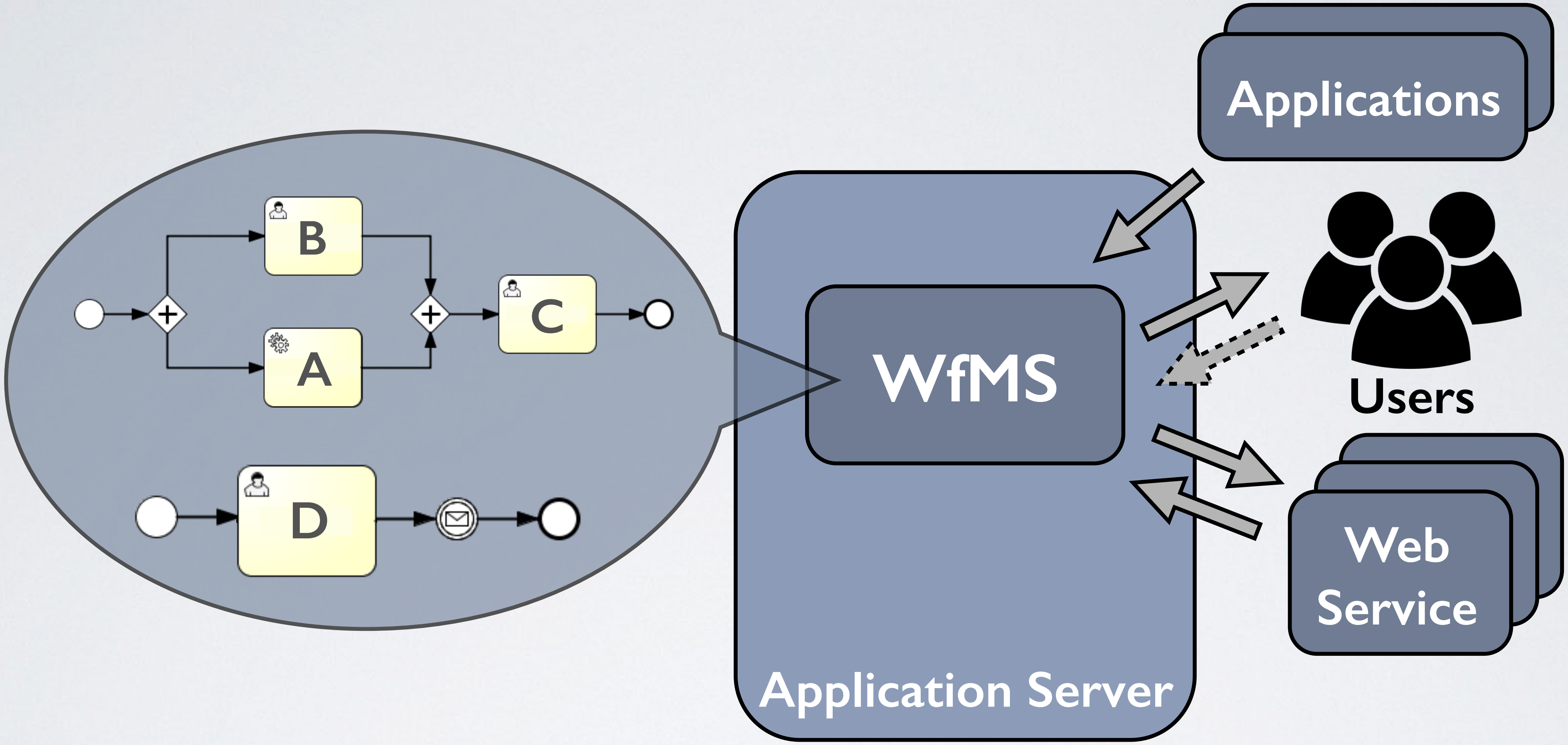
What is a Workflow Management System?



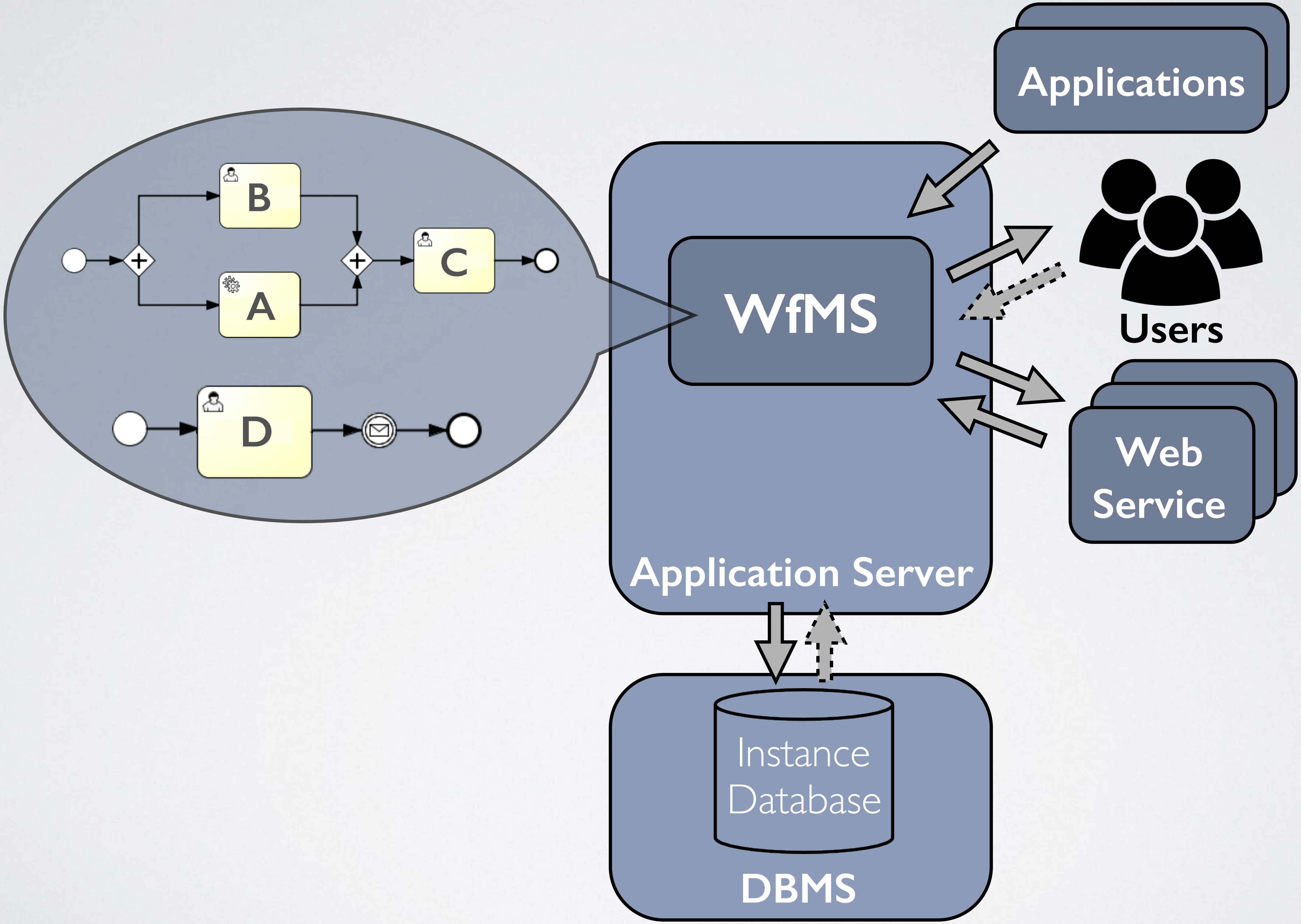
What is a Workflow Management System?



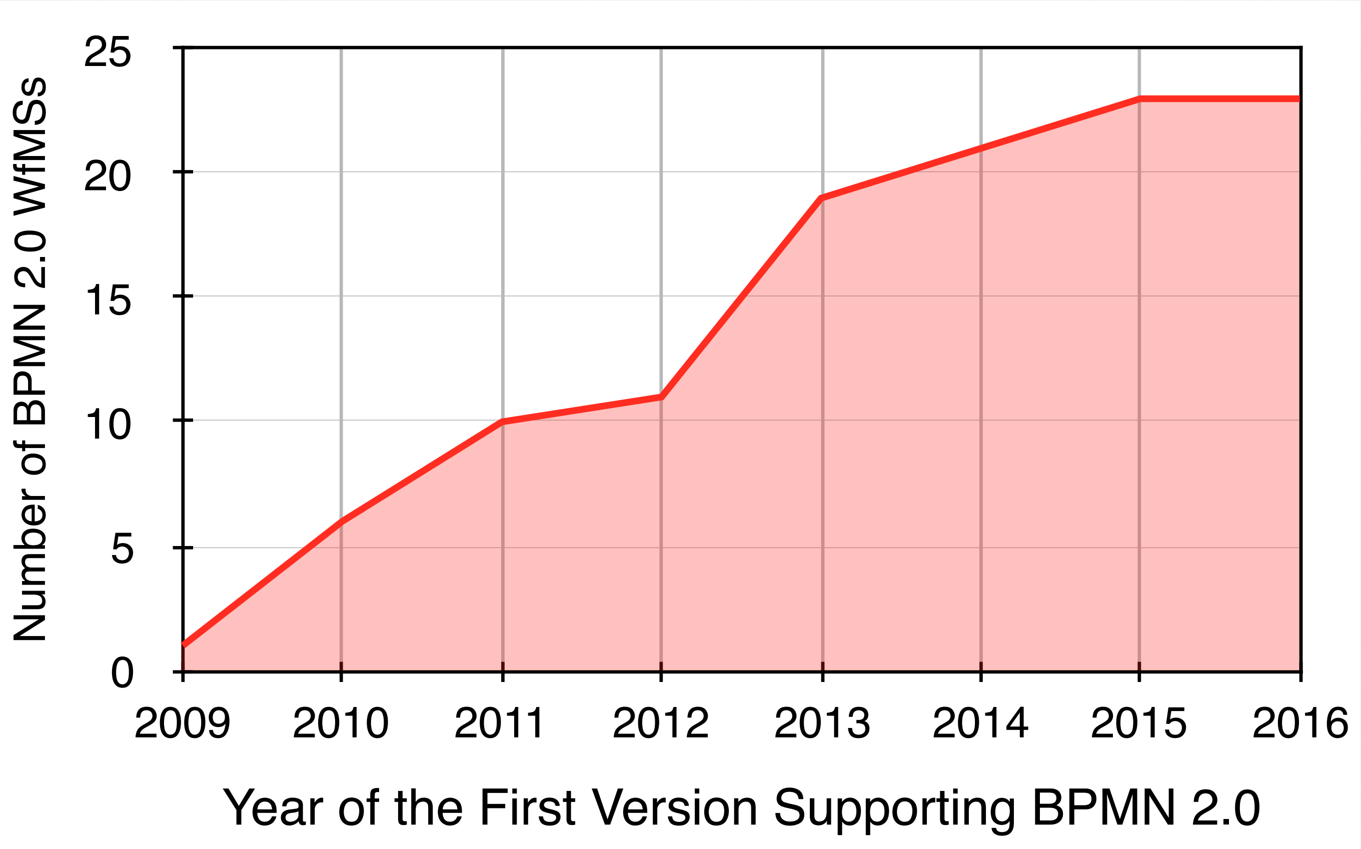
What is a Workflow Management System?



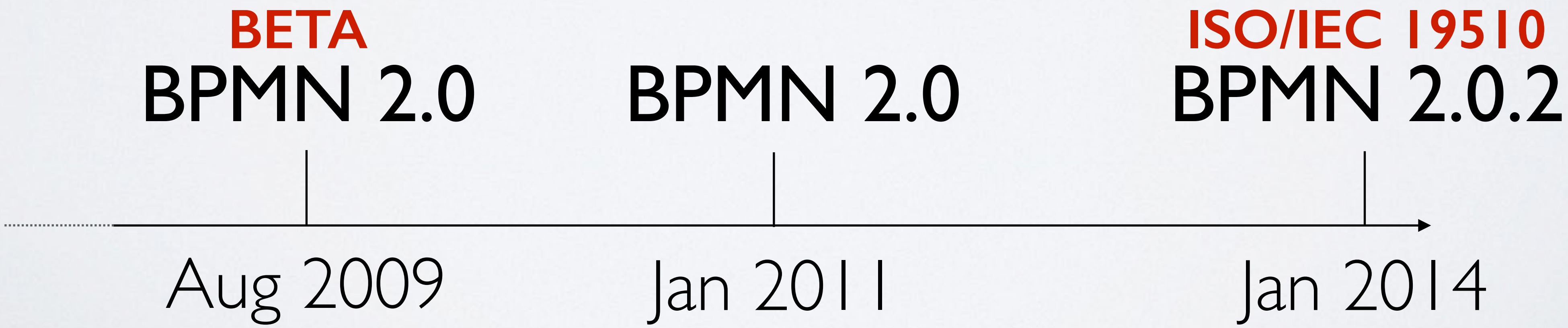
What is a Workflow Management System?



Many Vendors of BPMN 2.0 WfMSs



https://en.wikipedia.org/wiki/List_of_BPMN_2.0_engines



BETA
BPMN 2.0
Aug 2009

BPMN 2.0
Jan 2011

ISO/IEC 19510
BPMN 2.0.2
Jan 2014

Benchmarking Requirements

- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

- K. Huppler, **The art of building a good benchmark**, 2009
- J. Gray, **The Benchmark Handbook for Database and Transaction Systems**, 1993
- S. E. Sim, S. Easterbrook et al., **Using benchmarking to advance research: A challenge to software engineering**, 2003

Why a new Methodology?

No available methodologies involving vendors for both defining a standard benchmark and benchmarking production systems

Why a new Methodology?

No available methodologies involving vendors for both defining a standard benchmark and benchmarking production systems

spec TPC™

Already standard benchmarks

Why a new Methodology?

No available methodologies involving vendors for both defining a standard benchmark and benchmarking production systems

Research

Already standard benchmarks

No interaction with Vendors

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple

- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple

- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Lightweight

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple

- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Lightweight

Negligible Performance Impact

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple

- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Lightweight

Negligible Performance Impact

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Lightweight

Negligible Performance Impact

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Lightweight

Negligible Performance Impact

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Lightweight

Negligible Performance Impact

Why a Container-Centric Methodology?

- Relevant
- Representative
- Portable
- Scalable
- Simple

- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Emerging Technology

Lightweight

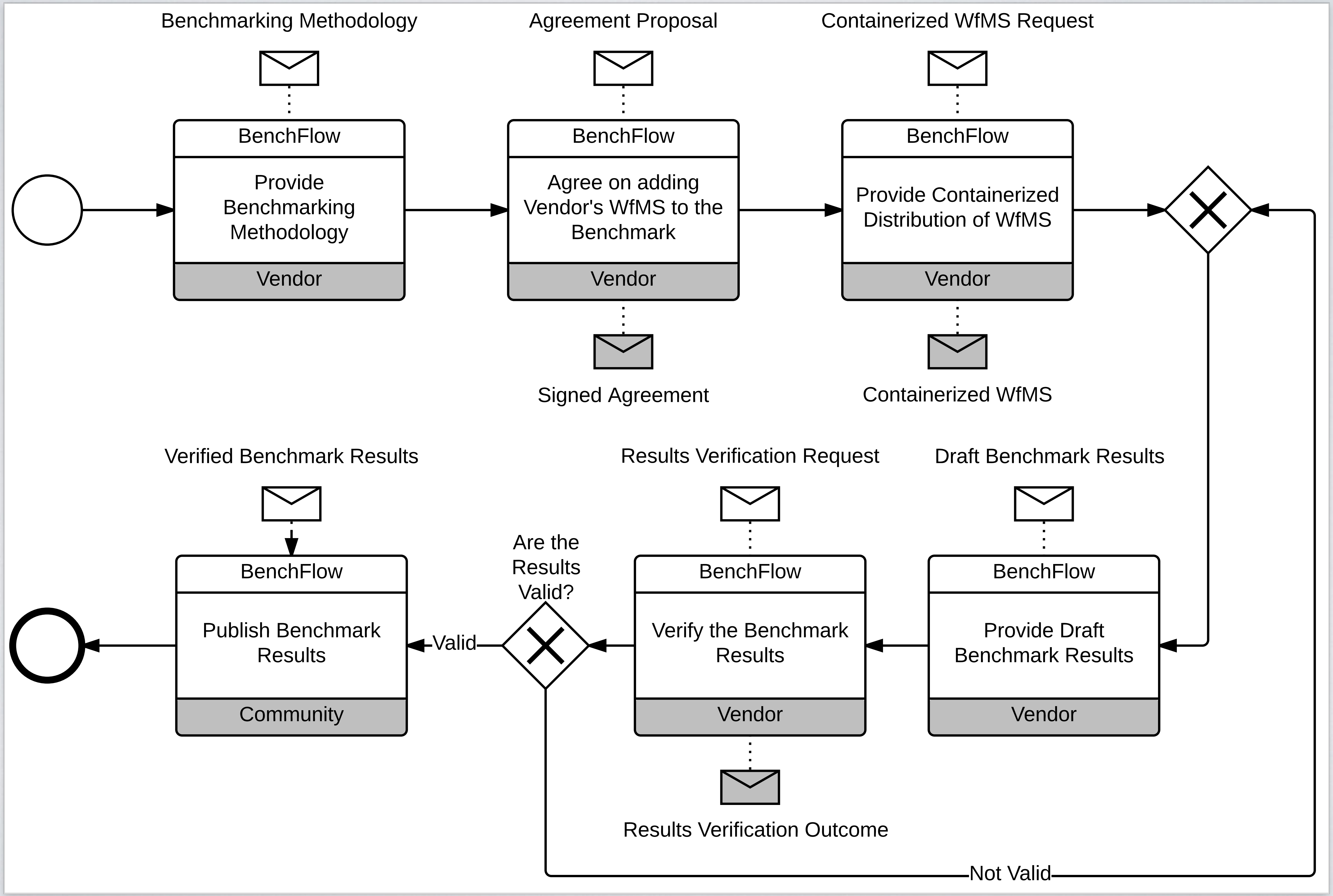
Negligible Performance Impact

What about the other Requirements?

- Relevant
- Representative
- Portable
- Scalable
- Simple

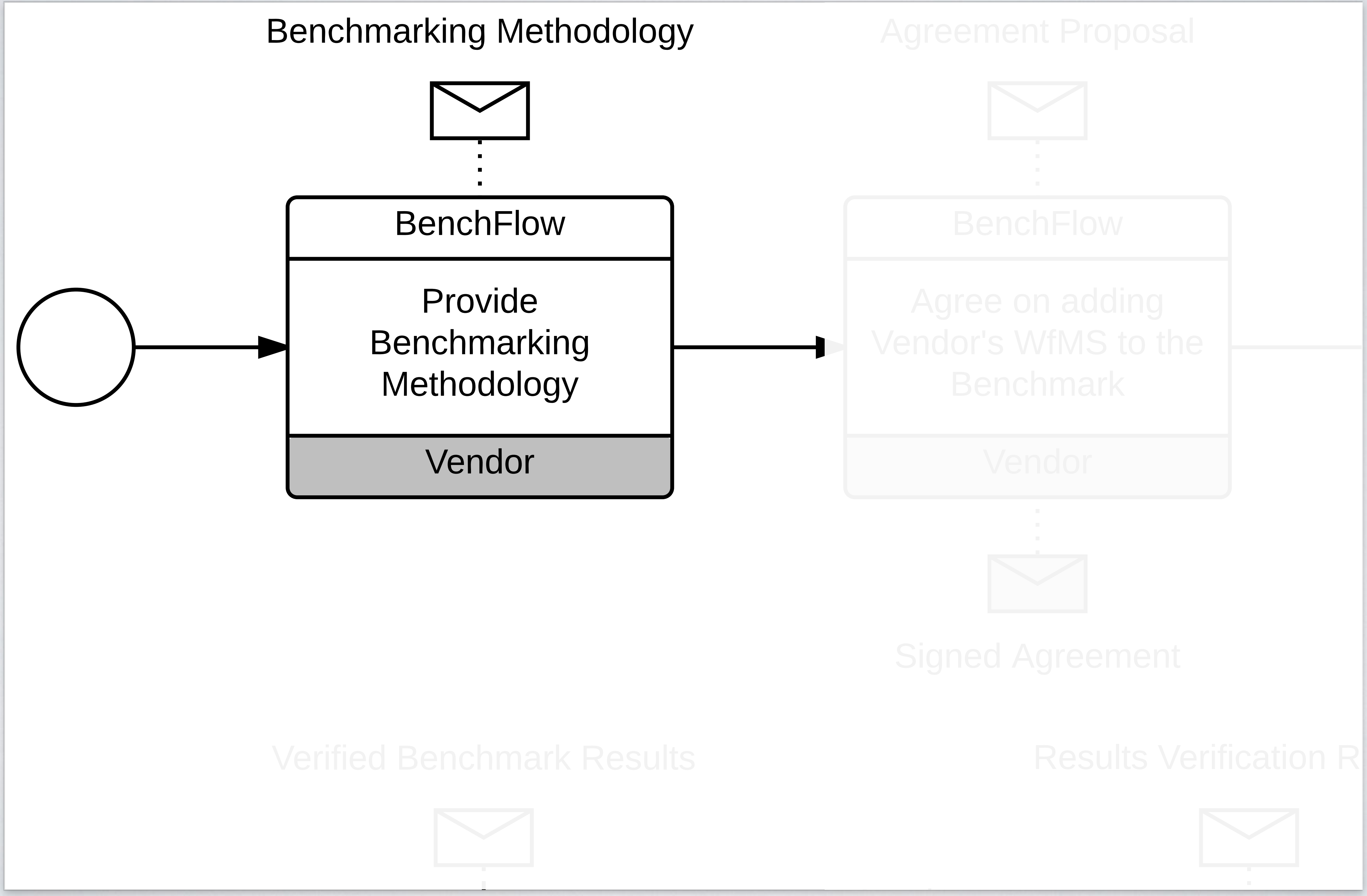
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

Benchmarking Choreography


Benchmarking Methodology
BenchFlow
Provide Benchmarking Methodology
Vendor
Agreement Proposal
BenchFlow
Agree on adding Vendor's WfMS to the Benchmark
Vendor
Signed Agreement
Containerized WfMS Request
BenchFlow
Provide Containerized Distribution of WfMS
Vendor
Containerized WfMS
Draft Benchmark Results
BenchFlow
Provide Draft Benchmark Results
Vendor
Results Verification Request
BenchFlow
Verify the Benchmark Results
Vendor
Results Verification Outcome
Are the Results Valid?
Valid
Not Valid
Verified Benchmark Results
BenchFlow
Publish Benchmark Results
Community

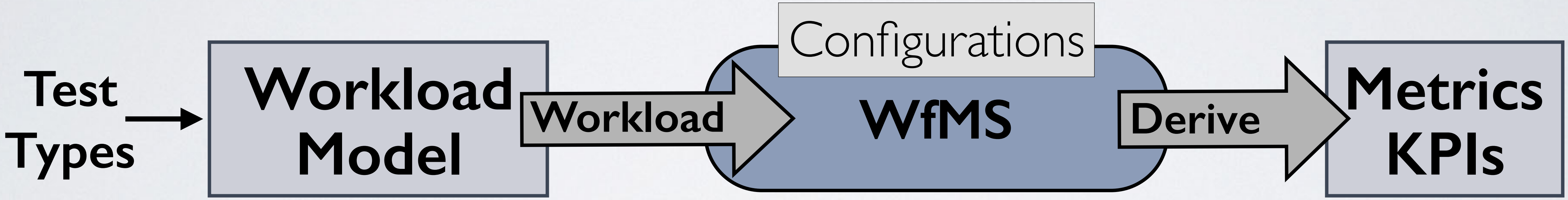
Benchmarking Methodology

provide the methodology



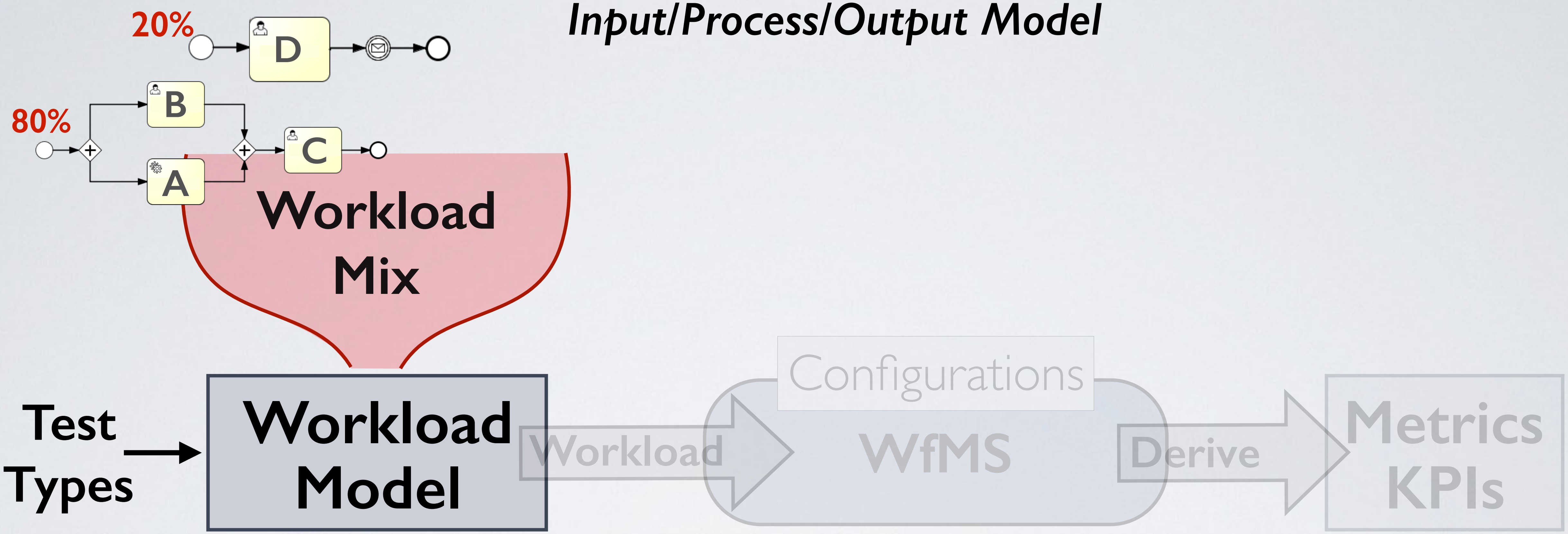
The Benchmarking Process

Input/Process/Output Model



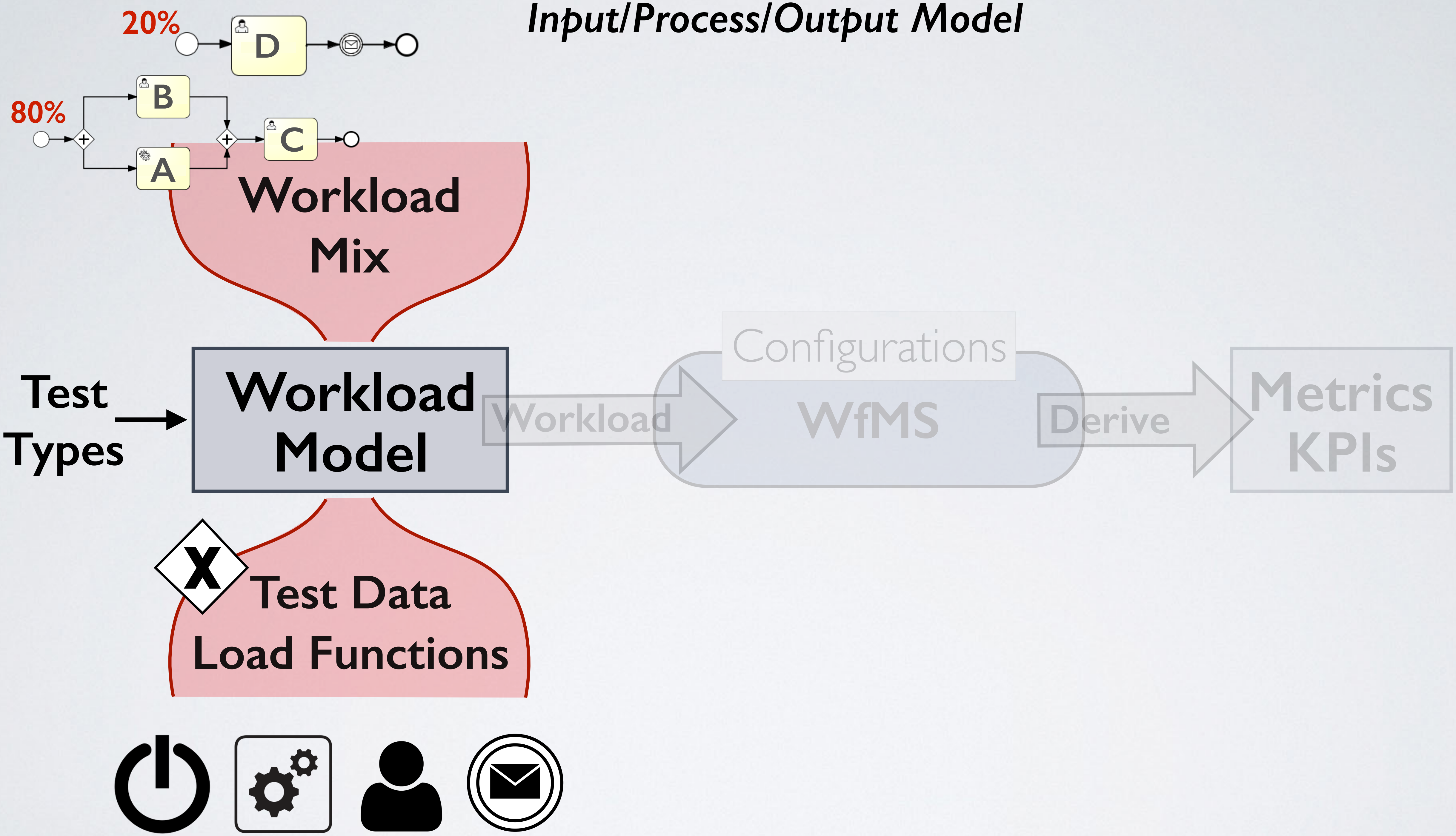
The Benchmarking Process

Input/Process/Output Model



The Benchmarking Process

Input/Process/Output Model



Workload Mix

Test Types

Workload Model

Test Data
Load Functions

Workload

Configurations

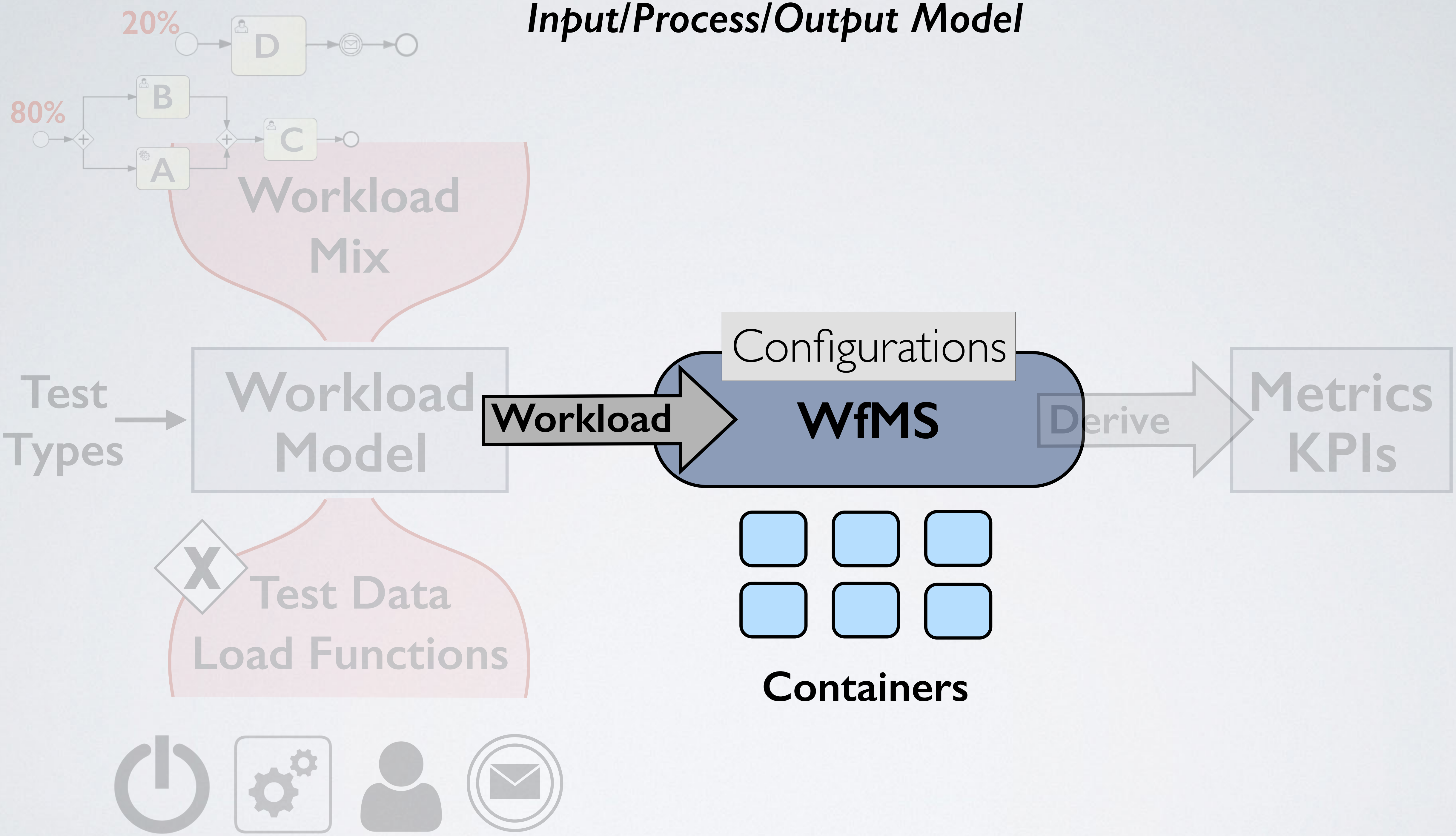
WfMS

Derive

Metrics
KPIs

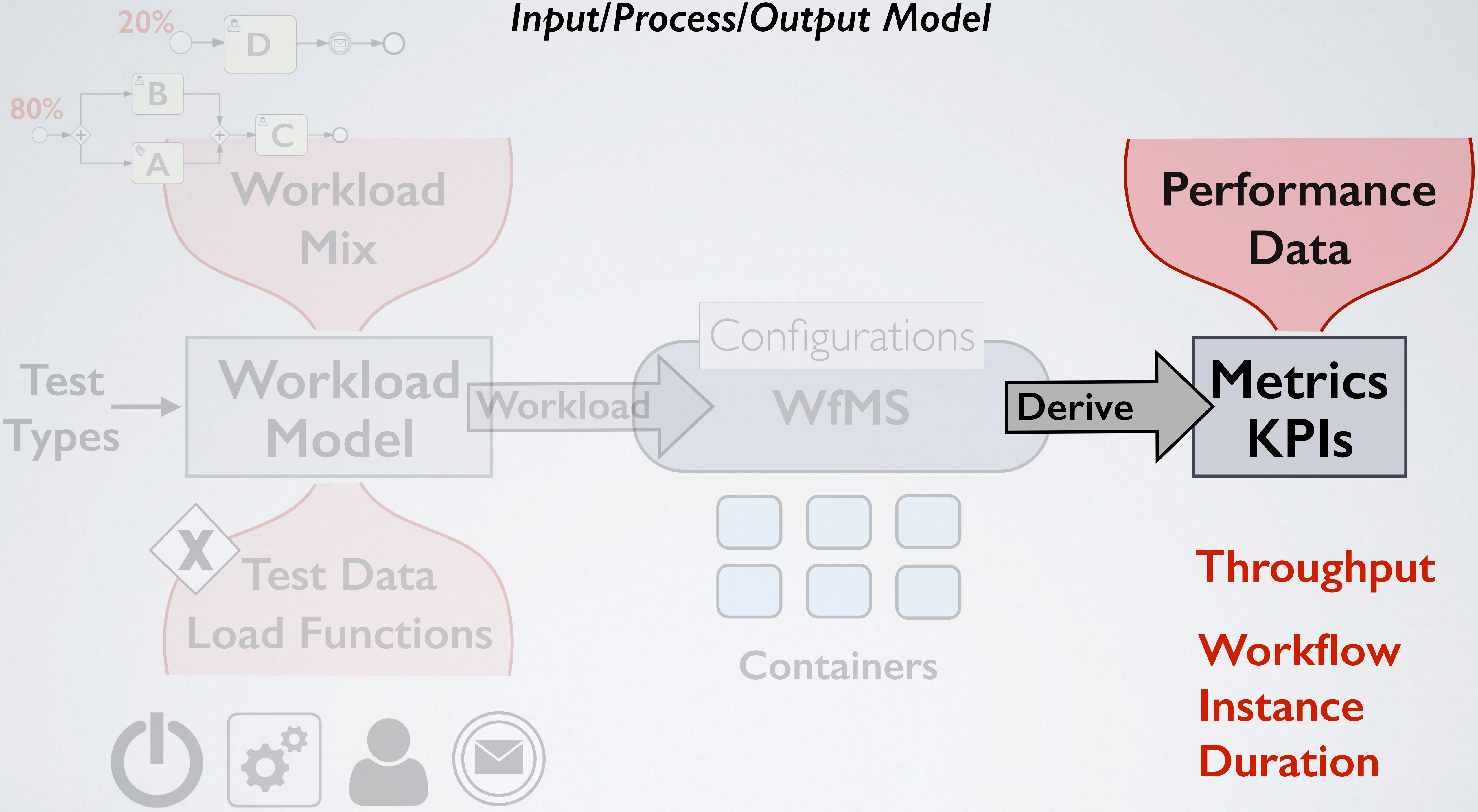
The Benchmarking Process

Input/Process/Output Model



The Benchmarking Process

Input/Process/Output Model



20%
D
80%
B
C
A
Workload Mix

Test Types
Workload Model
Workload
Configurations
WfMS
Derive
Performance Data
Metrics KPIs

X
Test Data
Load Functions

Containers

Throughput

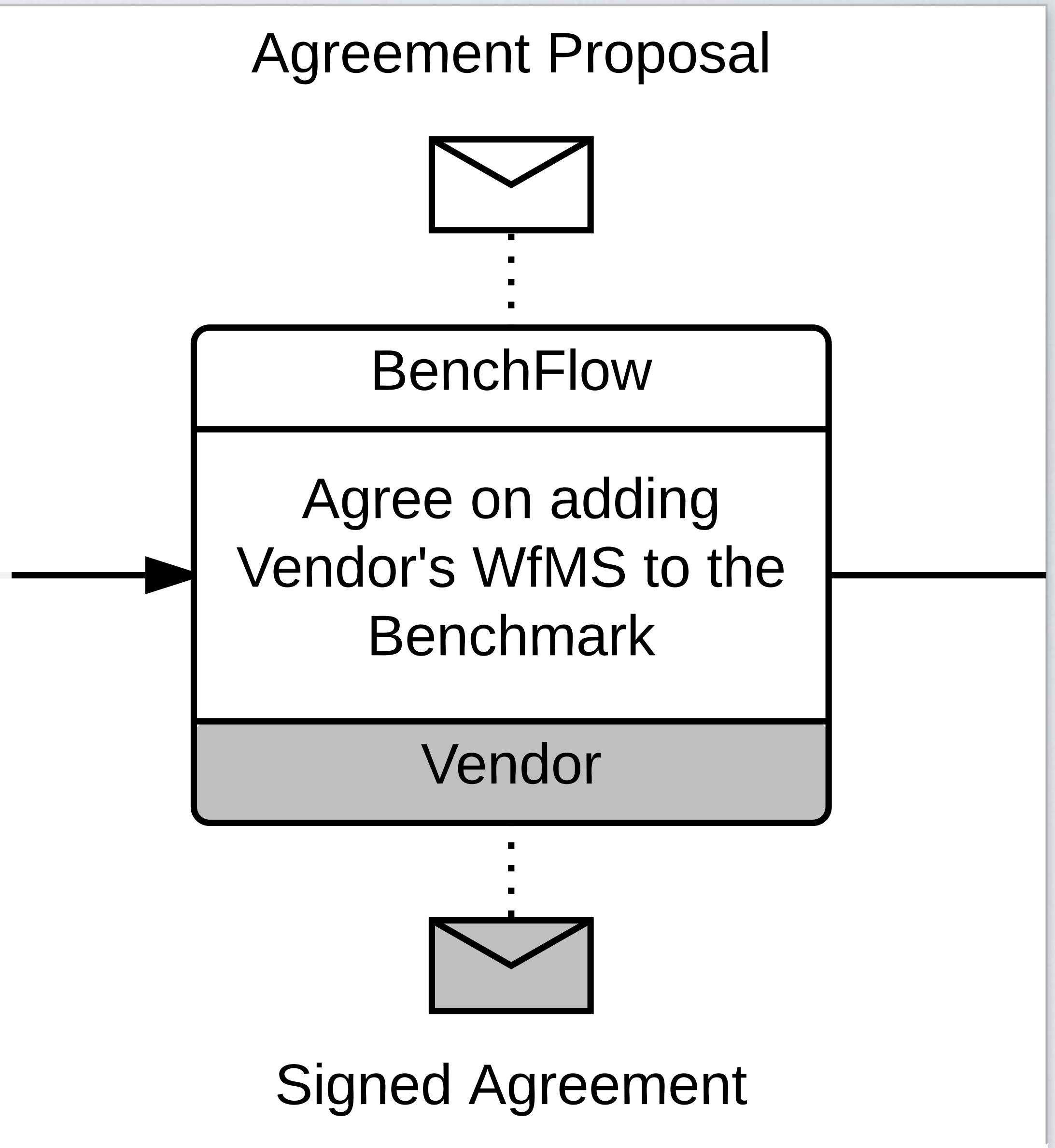
Workflow Instance Duration

Benchmarking Methodology

agreement with vendors

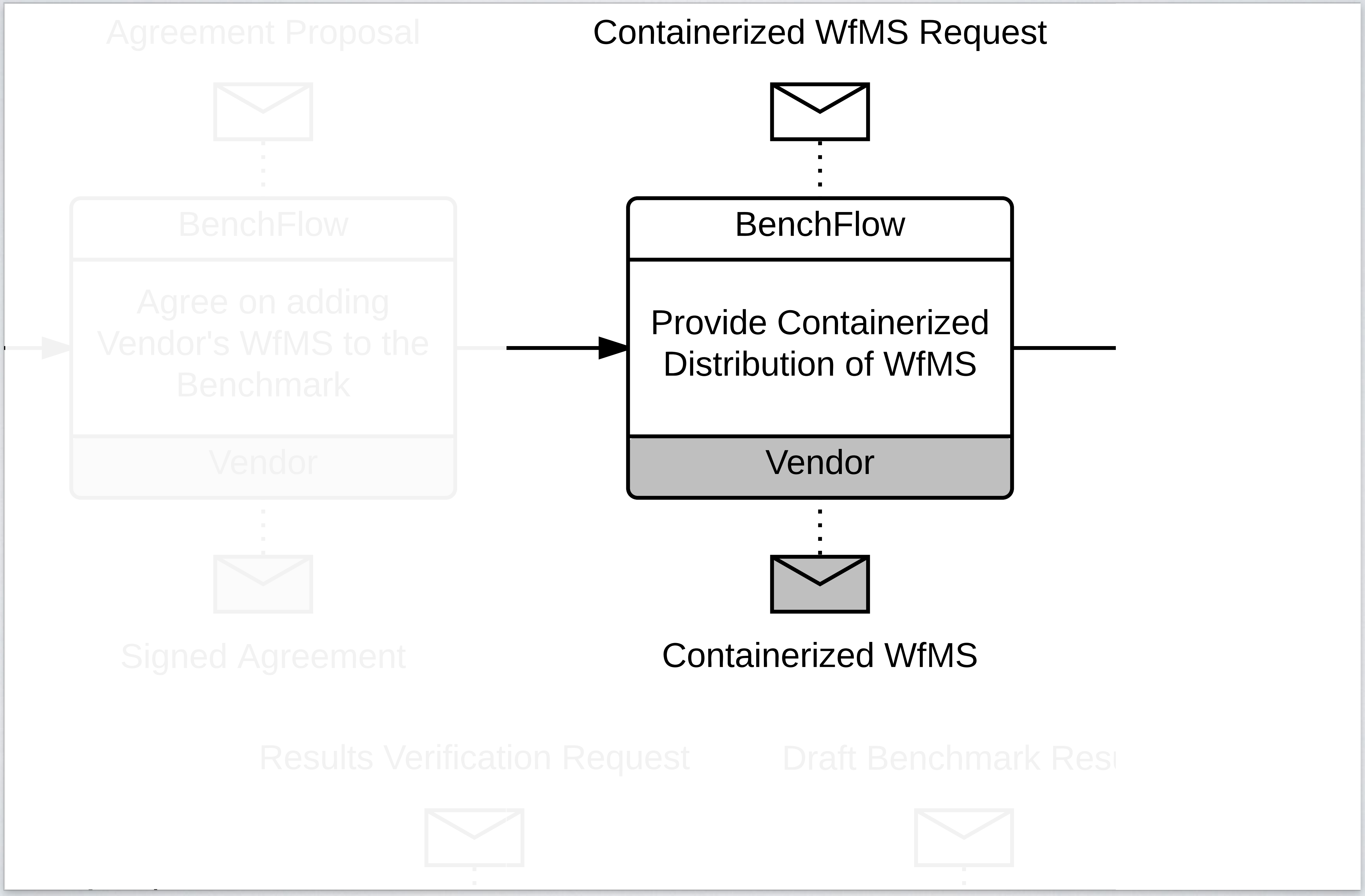
Main Agreement Points:

- Production Stable Release
- Provide defined APIs
- Share Containerised WfMS
- Authorise Publishing of Results



Benchmarking Methodology

containerised WfMSs


Agreement Proposal
BenchFlow
Agree on adding Vendor's WfMS to the Benchmark
Vendor
Signed Agreement
Containerized WfMS Request
BenchFlow
Provide Containerized Distribution of WfMS
Vendor
Containerized WfMS
Results Verification Request
Draft Benchmark Resu

Configurations
WfMS

Benchmarking Methodology

requirements from the WfMS

CORE

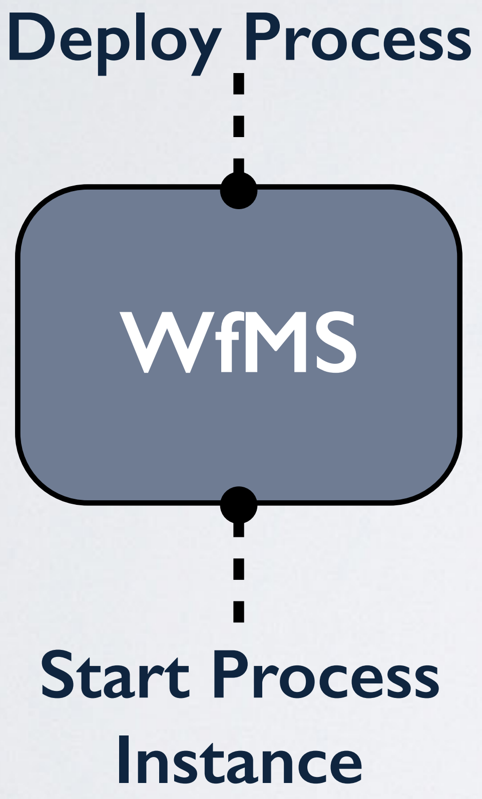
NON-CORE

Configurations
WfMS

Benchmarking Methodology

requirements from the WfMS

Initialisation APIs



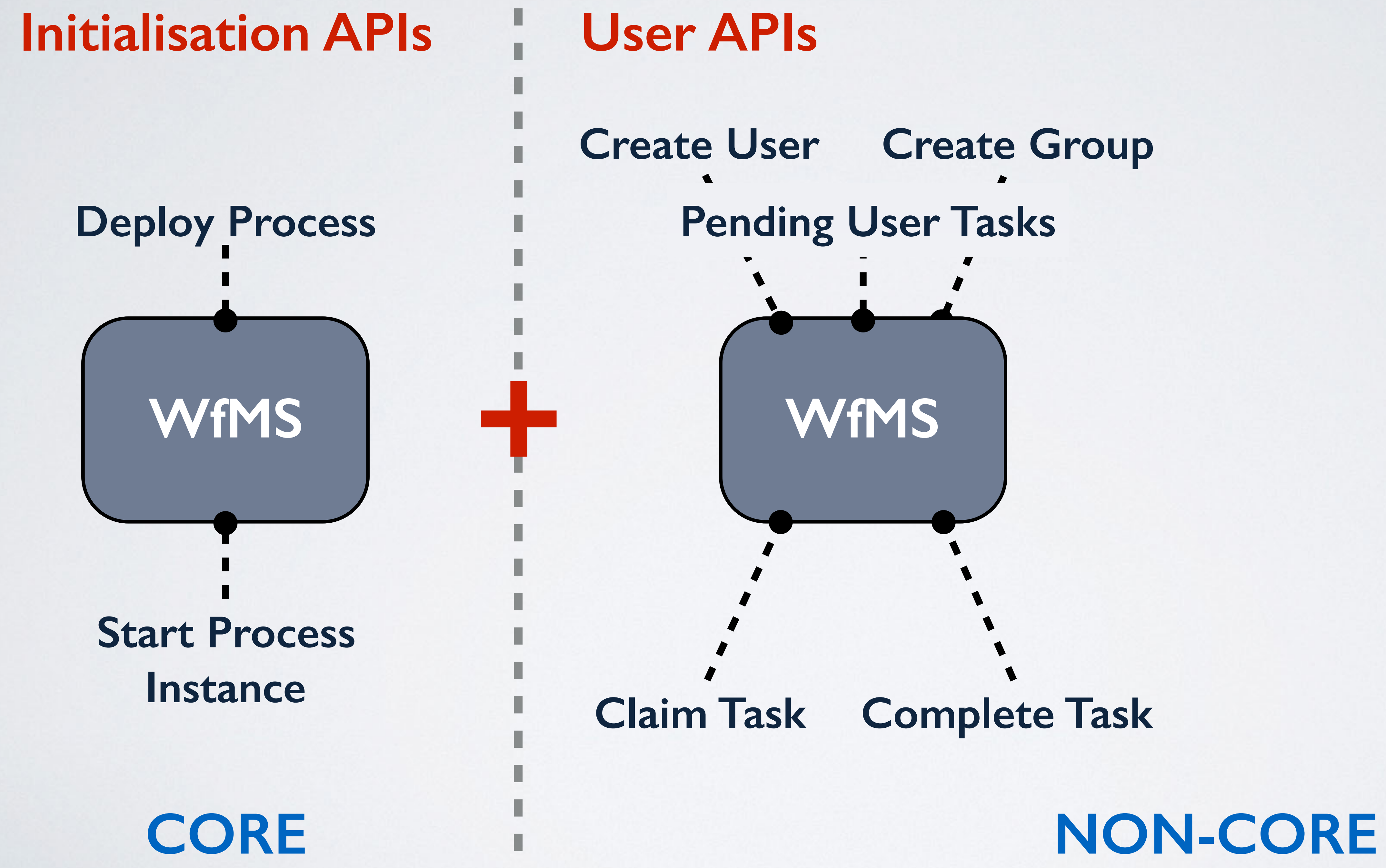
CORE

NON-CORE

Configurations
WfMS

Benchmarking Methodology

requirements from the WfMS

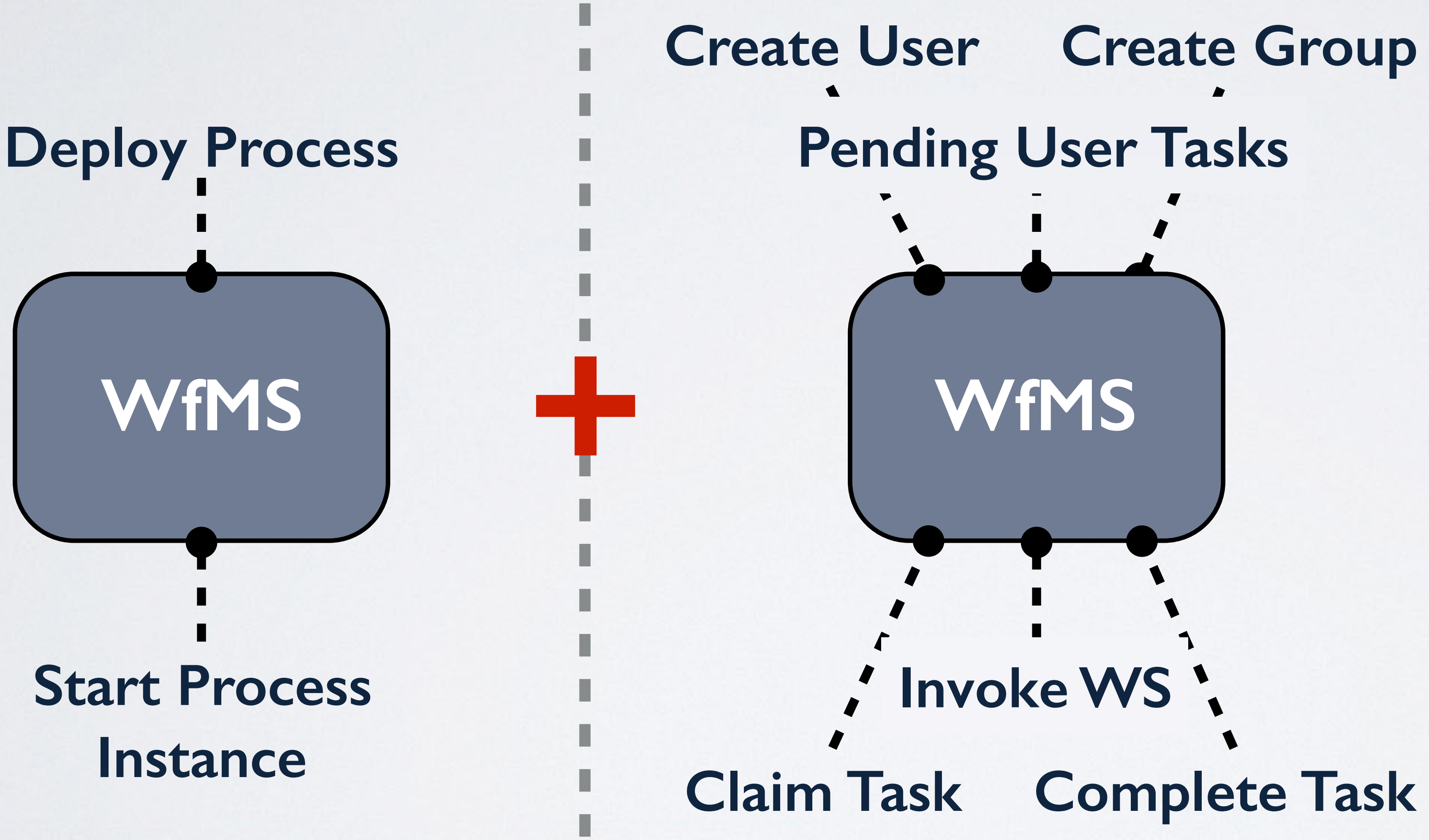


Configurations
WfMS

Benchmarking Methodology

requirements from the WfMS

Initialisation APIs | User APIs Web Service APIs



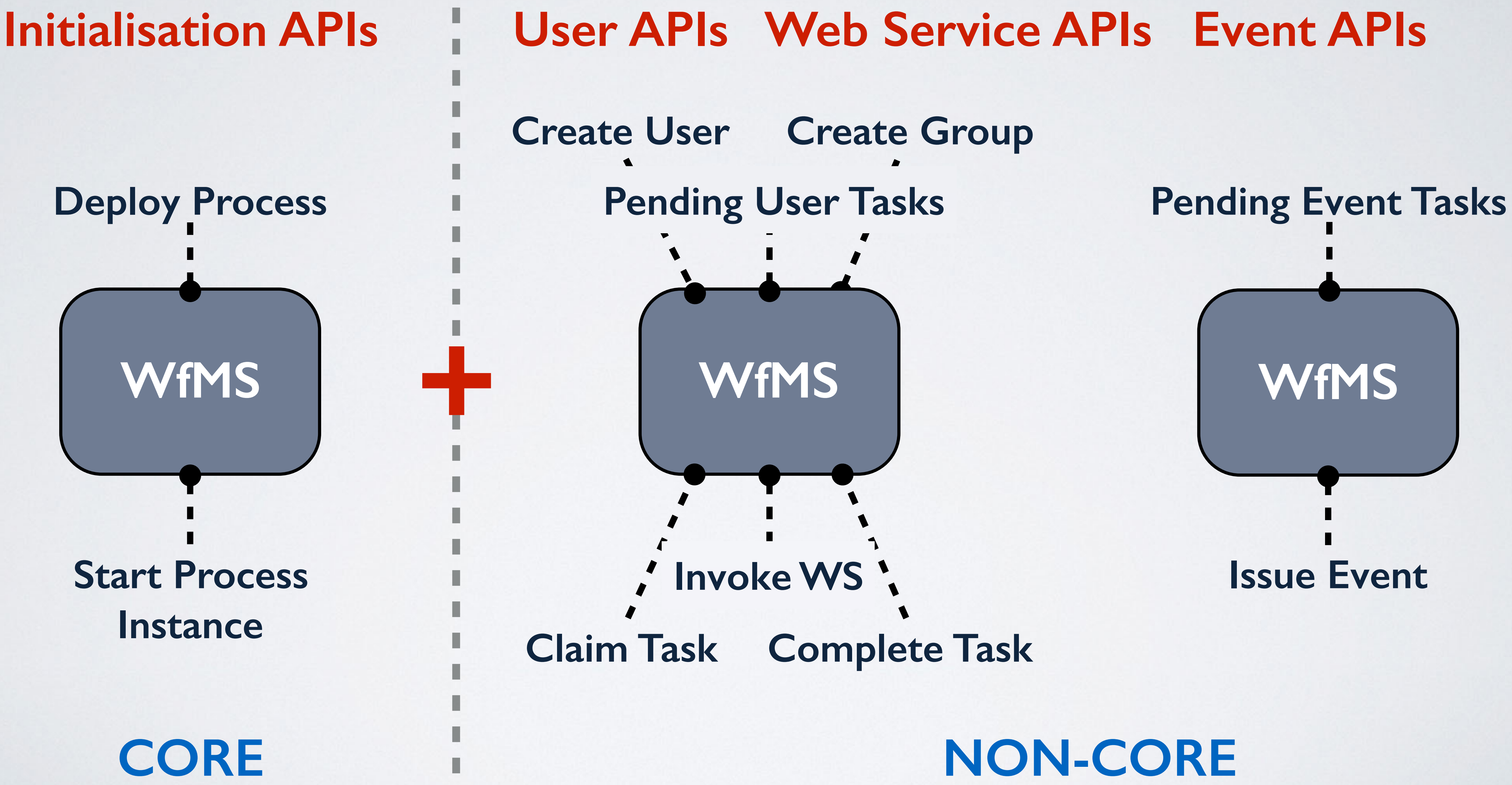
CORE

NON-CORE

Configurations
WfMS

Benchmarking Methodology

requirements from the WfMS



Initialisation APIs

Deploy Process

WfMS

Start Process Instance

CORE

User APIs

Create User

Create Group

Pending User Tasks

WfMS

Invoke WS

Claim Task

Complete Task

Web Service APIs

Event APIs

Pending Event Tasks

WfMS

Issue Event

NON-CORE

Configurations
WfMS

Benchmarking Methodology

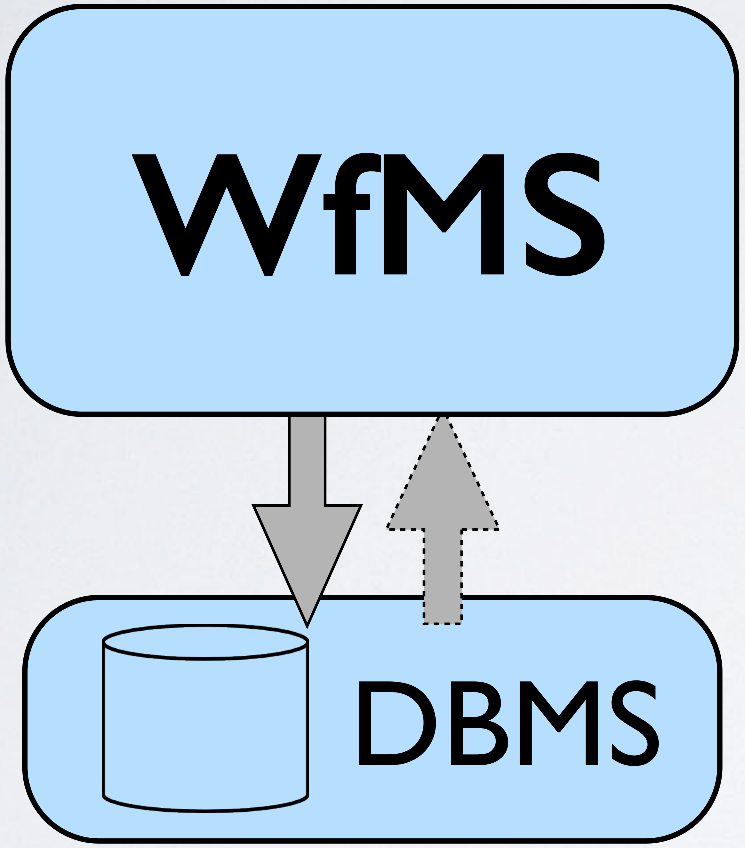
requirements from the WfMS

		Functionality	**Min Response Data**
Core APIs	**Initialisation APIs**	Deploy a process	Deployed process ID
		Start a process instance	Process instance ID
Non-core APIs	**User APIs**	Create a user	User ID
		Create a group of users	User group ID
		Access pending tasks	Pending tasks IDs
		Claim a task*	
		Complete a task	
	Event APIs	Access pending events	Pending events IDs
		Issue events	
	Web service APIs	Map tasks to Web service endpoints	

*Optional depending on the WfMS implementation

Benchmarking Methodology

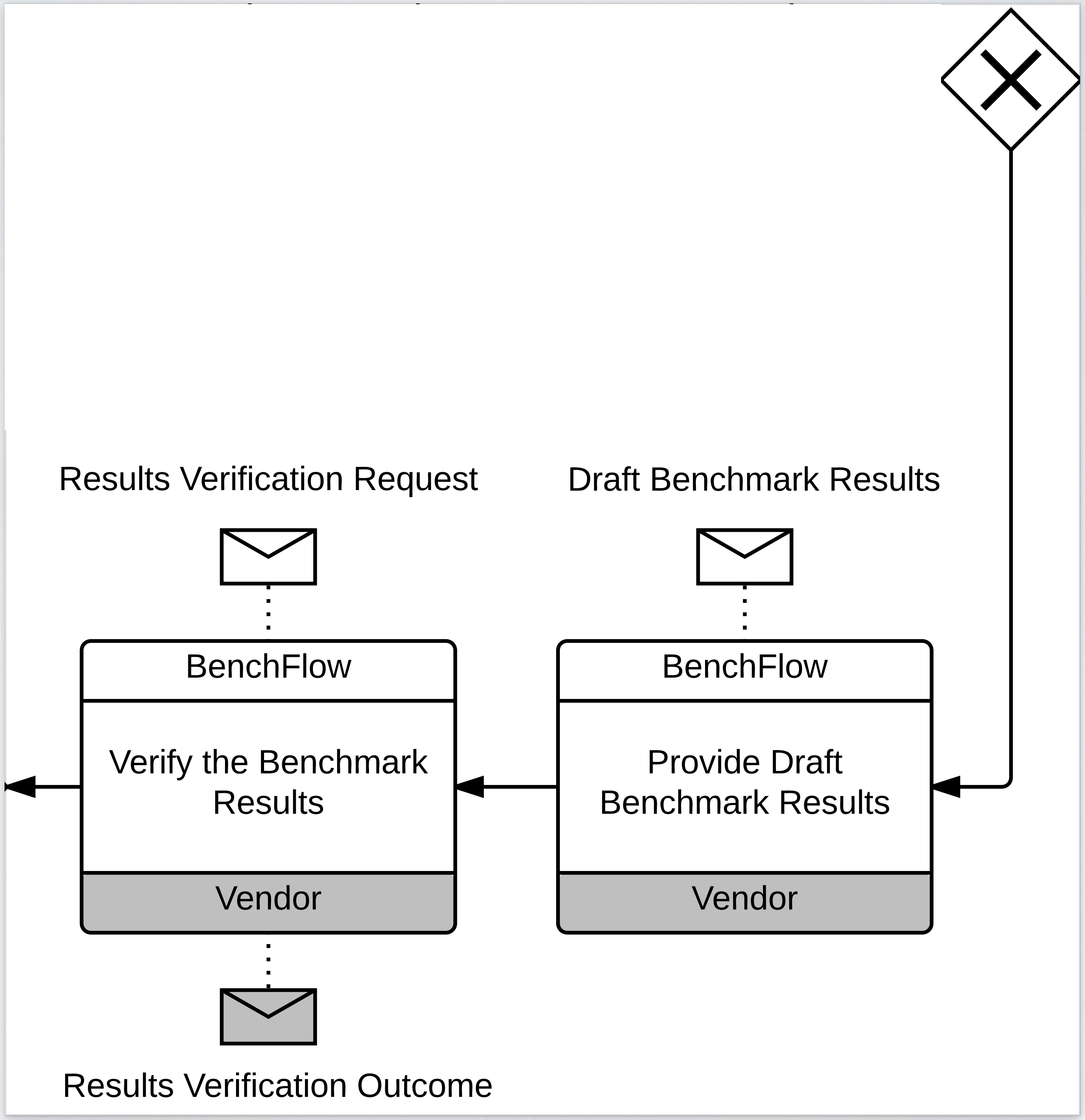
containerised WfMSs



- At least two containers
- DBMS can refer to existing one publicly available
- Provide a ready to use default configuration (at least)
- Configurability of: DBMS, WfMS, Logging Level (at least)

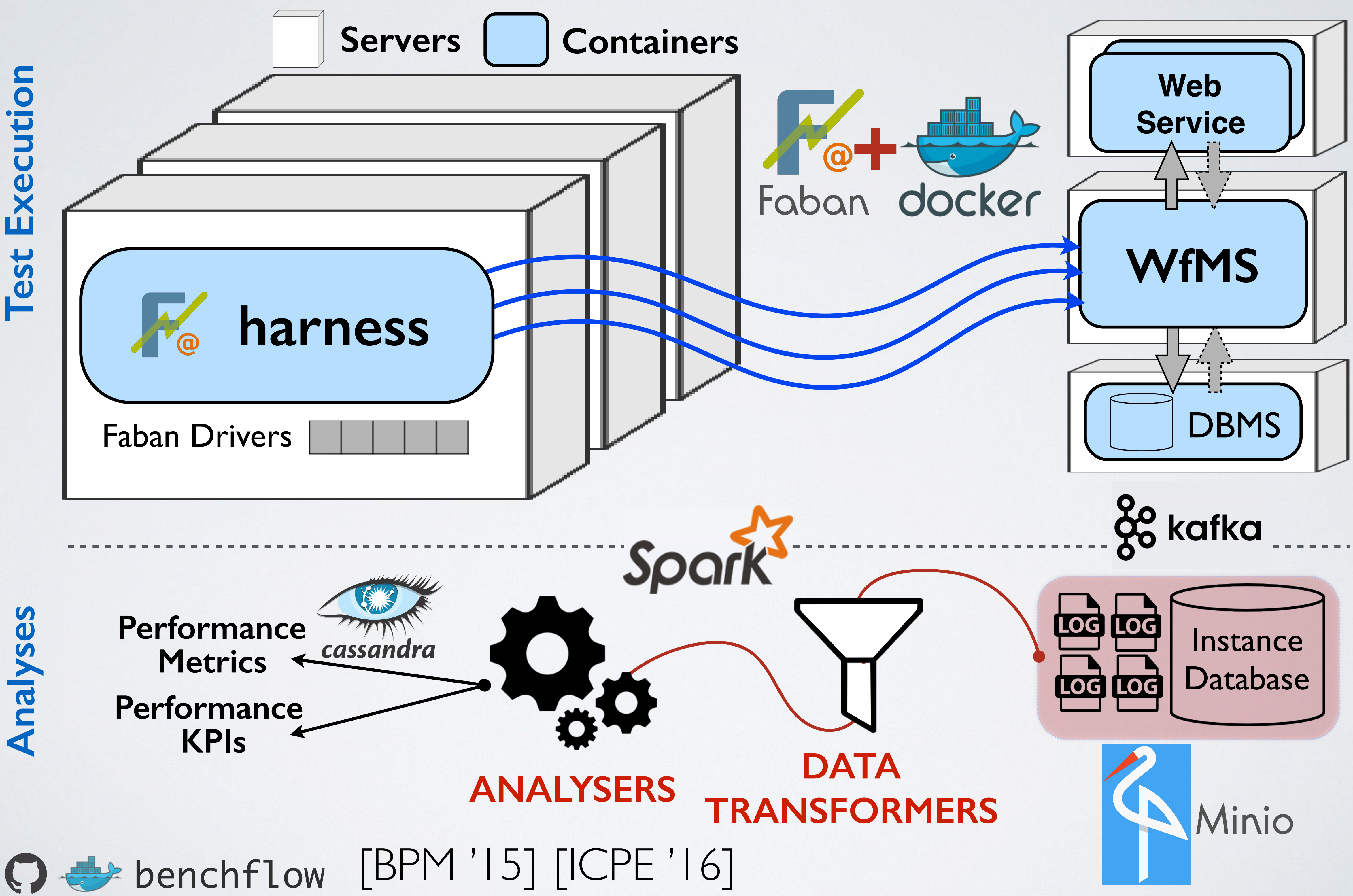
Benchmarking Methodology

executing the benchmark and providing results


Results Verification Request
Draft Benchmark Results
BenchFlow
Verify the Benchmark Results
Vendor
BenchFlow
Provide Draft Benchmark Results
Vendor
Results Verification Outcome

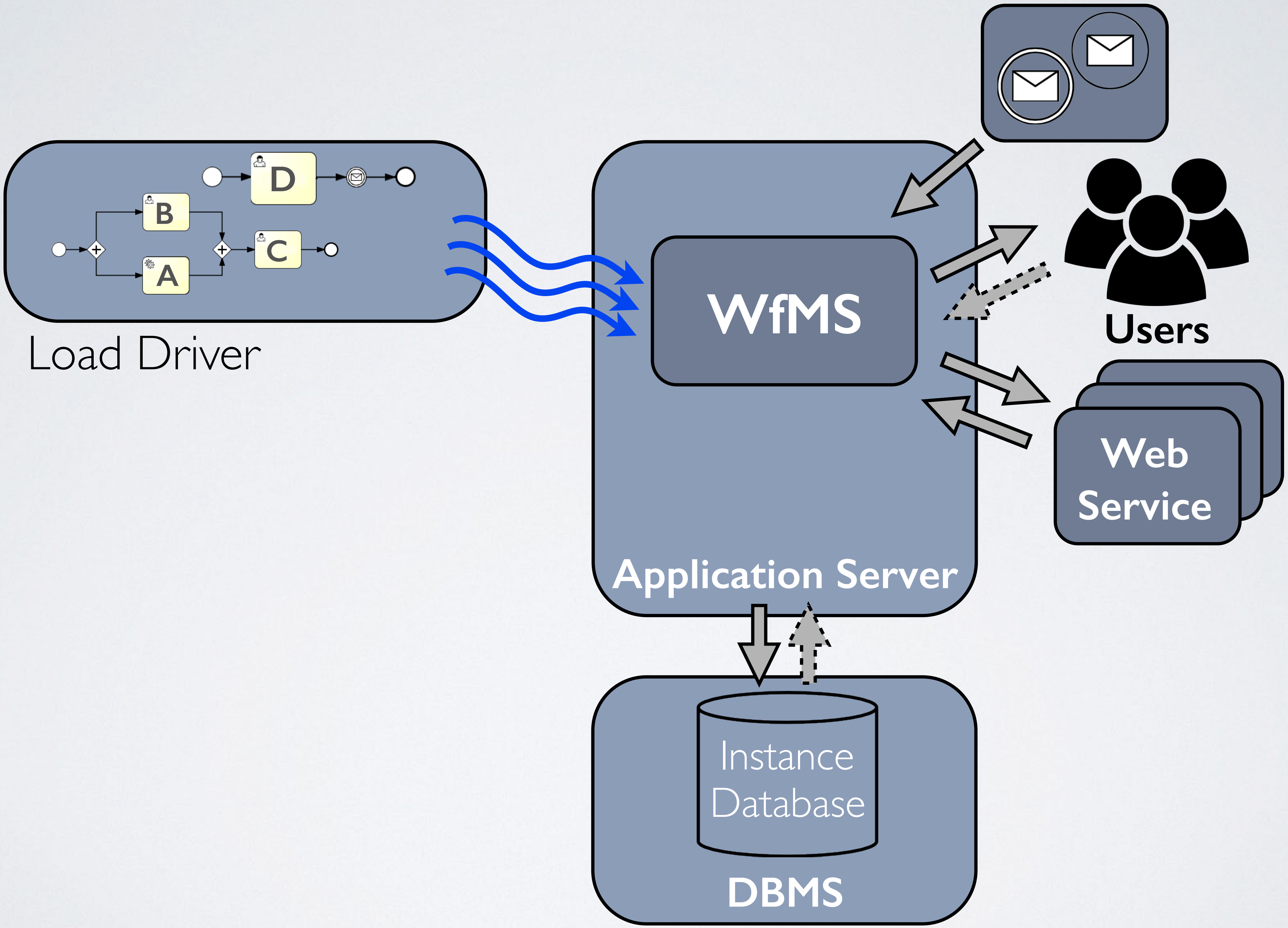
BenchFlow Framework

architecture

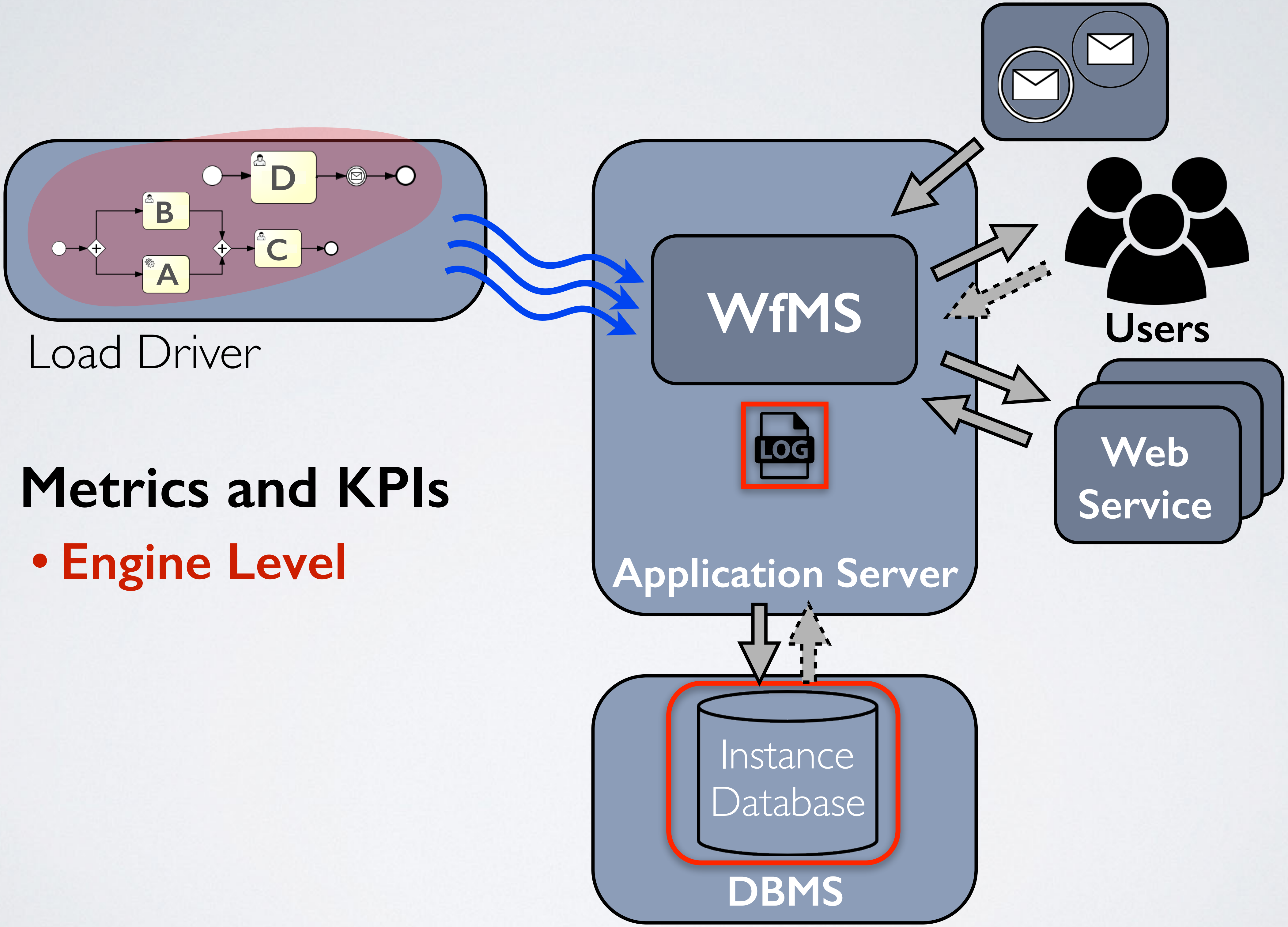


[BPM '15] [ICPE '16]

Performance Metrics and KPIs



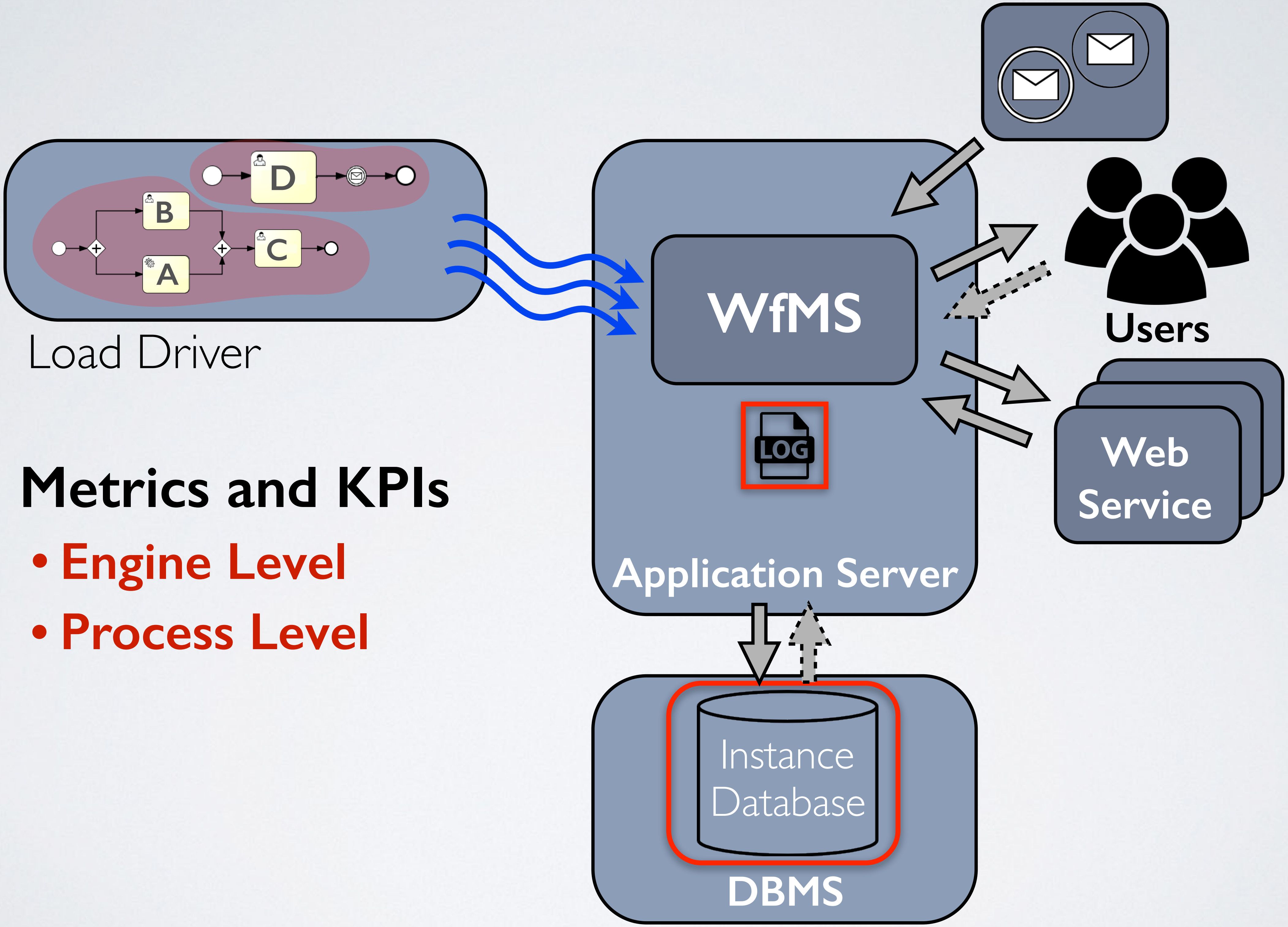
Performance Metrics and KPIs



Metrics and KPIs

- Engine Level

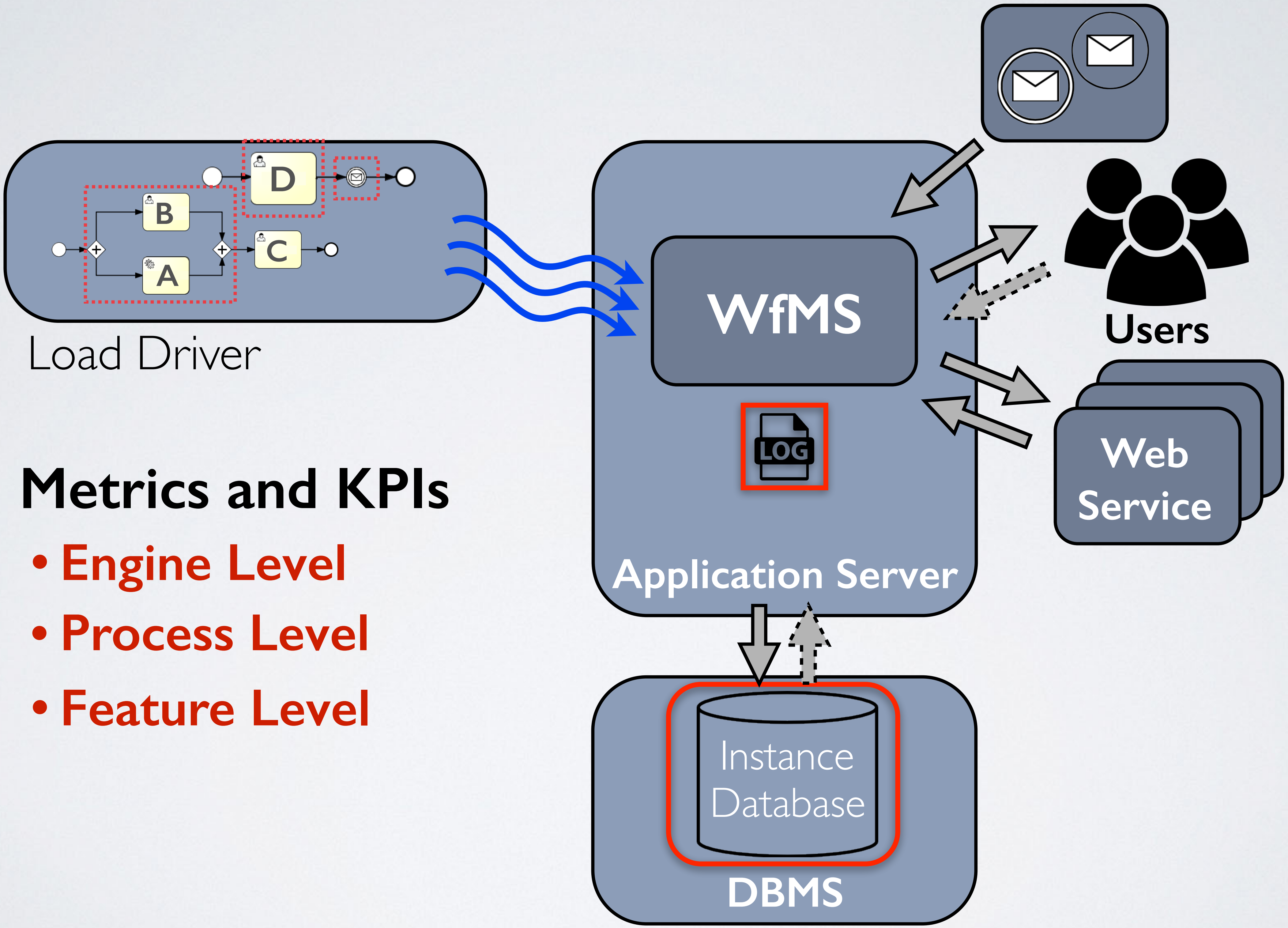
Performance Metrics and KPIs



Metrics and KPIs

- Engine Level
- Process Level

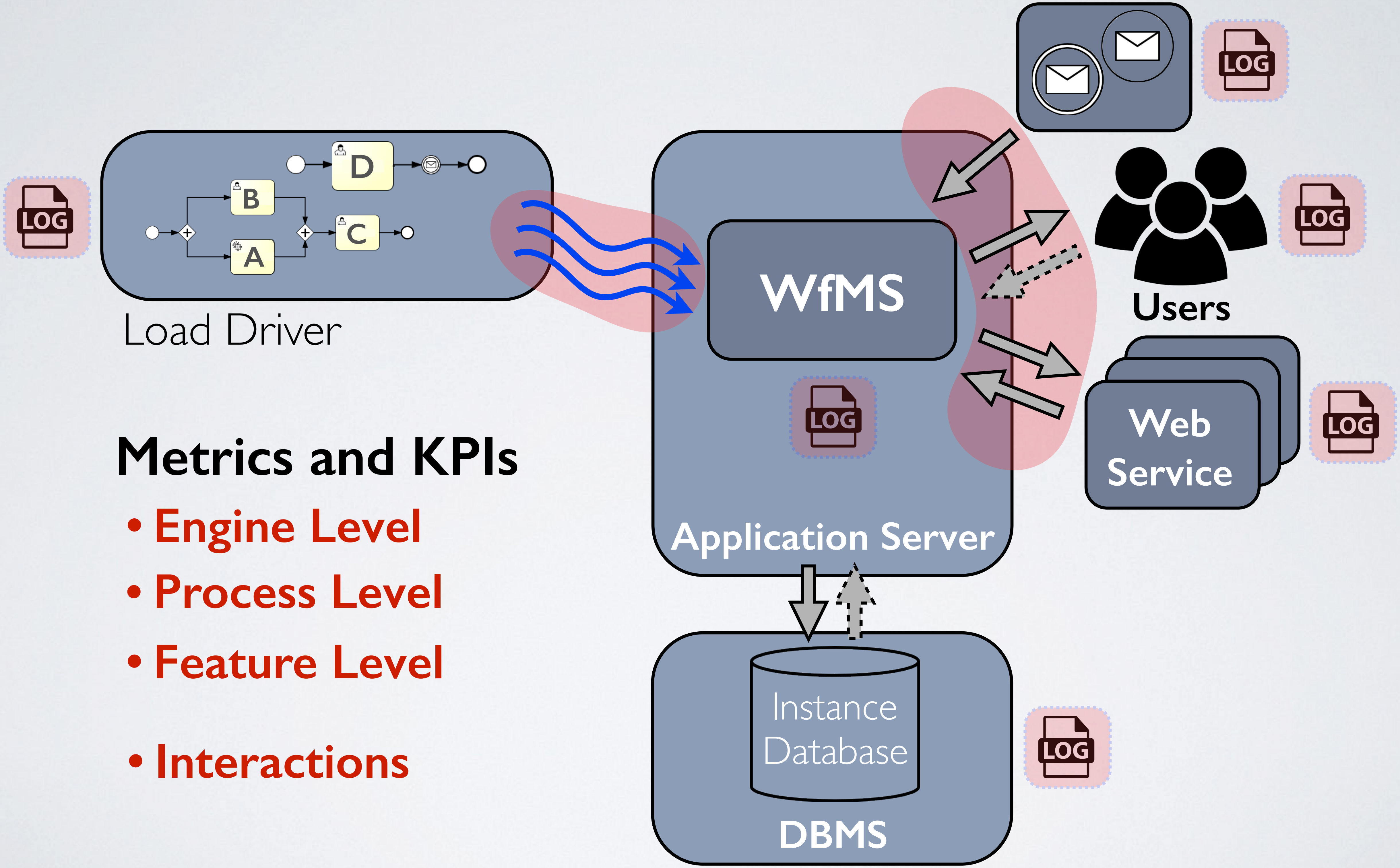
Performance Metrics and KPIs



Metrics and KPIs

- Engine Level
- Process Level
- Feature Level

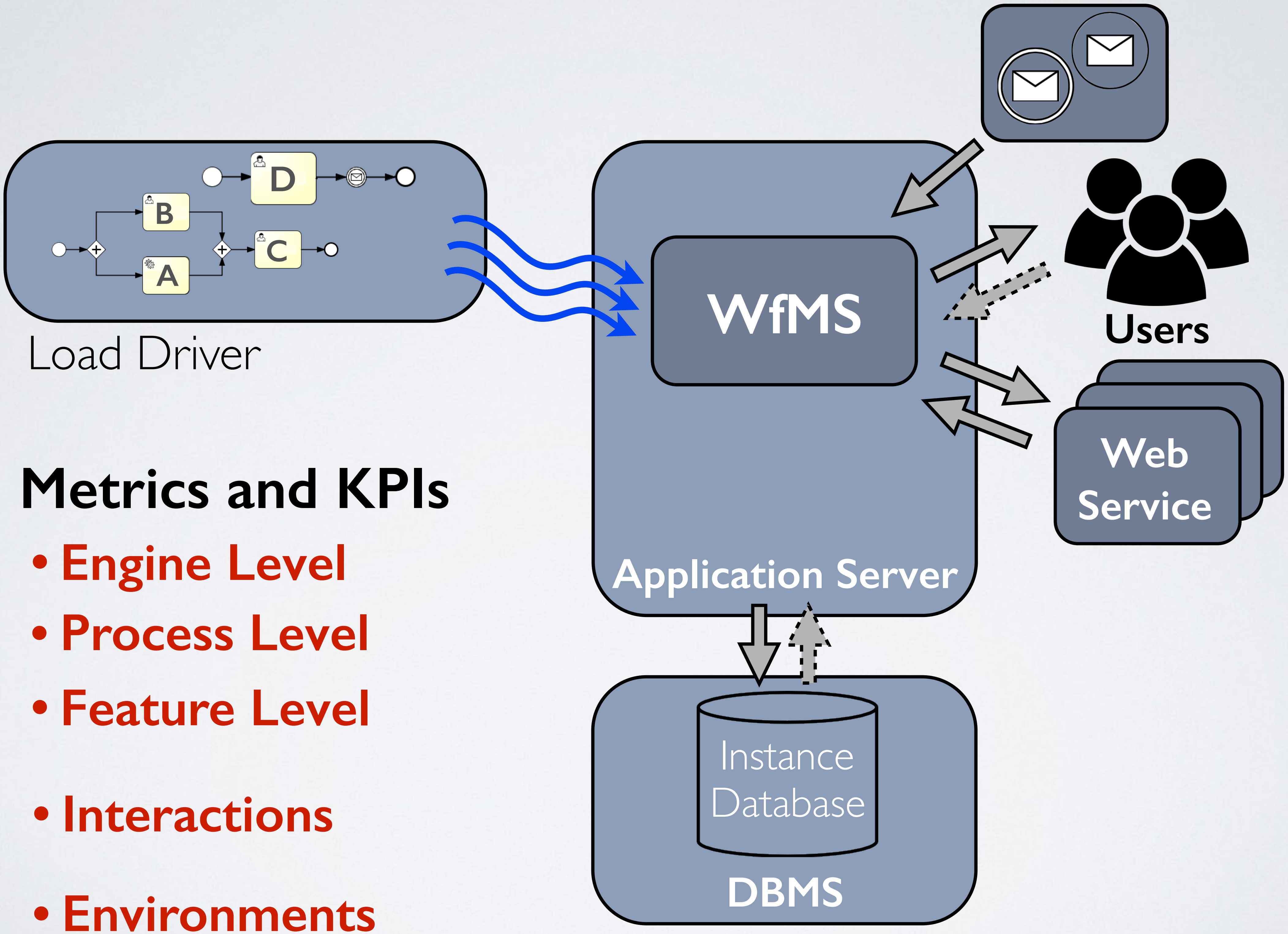
Performance Metrics and KPIs



Metrics and KPIs

- Engine Level
- Process Level
- Feature Level
- Interactions

Performance Metrics and KPIs



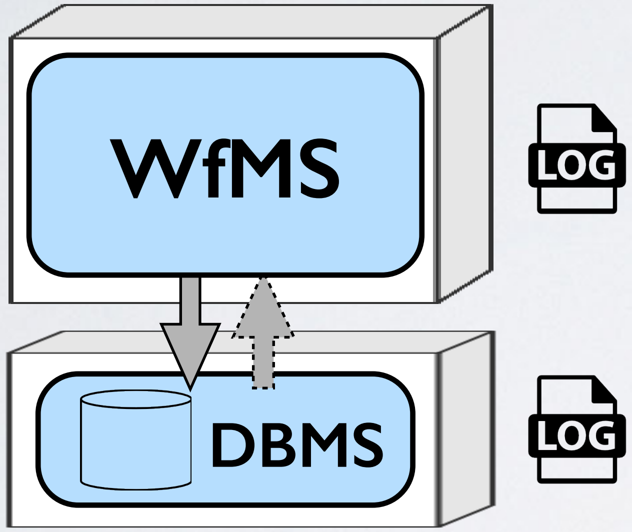
Metrics and KPIs

- Engine Level
- Process Level
- Feature Level
- Interactions
- Environments

Executing the Benchmark

minimal data requirements

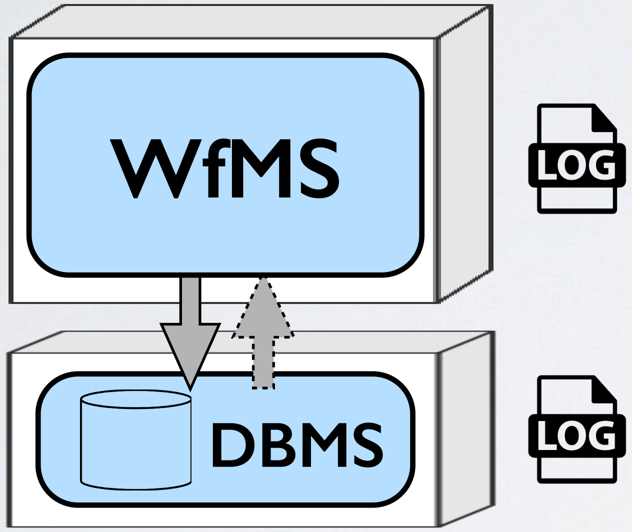
Accessibility of the Data



Executing the Benchmark

minimal data requirements

Accessibility of the Data



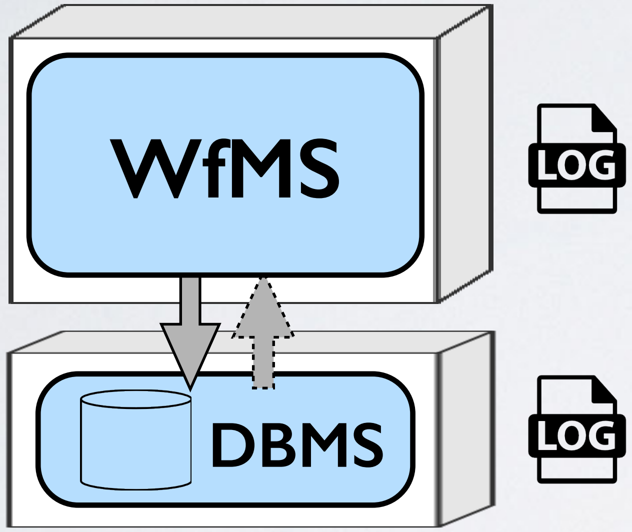
Availability of Timing Data

- Workflow & Construct:
  - Start Time
  - End Time
  - [Duration]

Executing the Benchmark

minimal data requirements

Accessibility of the Data



Availability of Timing Data

- Workflow & Construct:
  - Start Time
  - End Time
  - [Duration]

Availability of Execution State

State of the workflow execution. E.g., running, completed, error

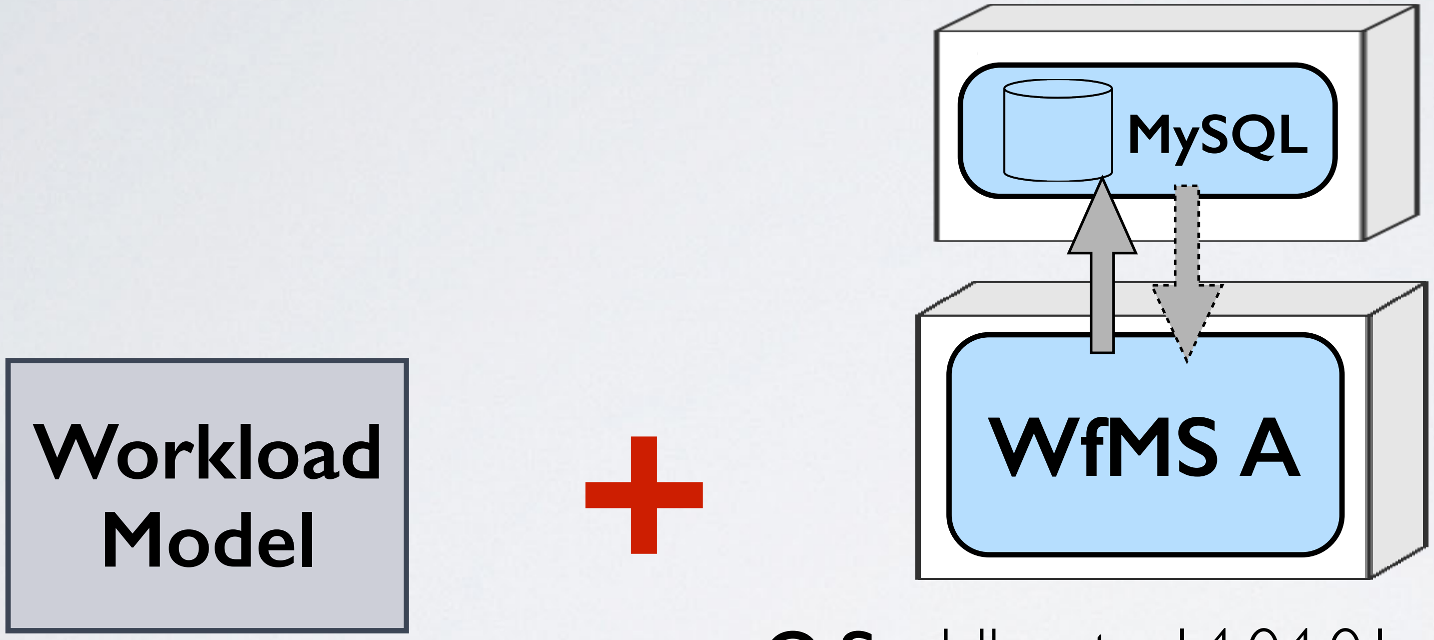
Benchmarking Methodology

mock example of benchmark results

Workload Model

Benchmarking Methodology

mock example of benchmark results



O.S.: Ubuntu 14.04.01
J.V.M.: Oracle Serv. 7u79
App. Server: Ap.Tomcat 7.0.62

MySQL: Community Server 5.6.26

WfMS A: v7.0.1

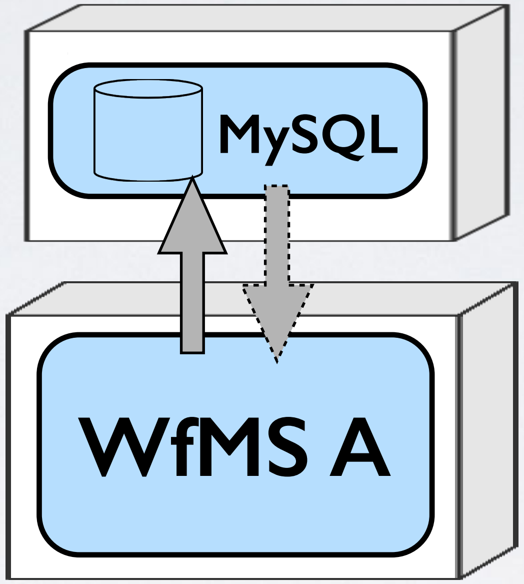
Hardware Configuration

Benchmarking Methodology

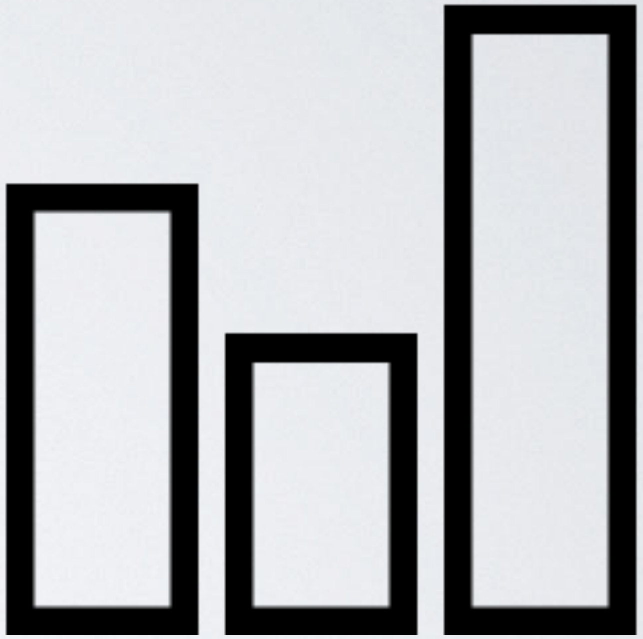
mock example of benchmark results

Workload Model

+



+



Metrics and KPIs

O.S.: Ubuntu 14.04.01
J.V.M.: Oracle Serv. 7u79
App. Server: Ap.Tomcat 7.0.62

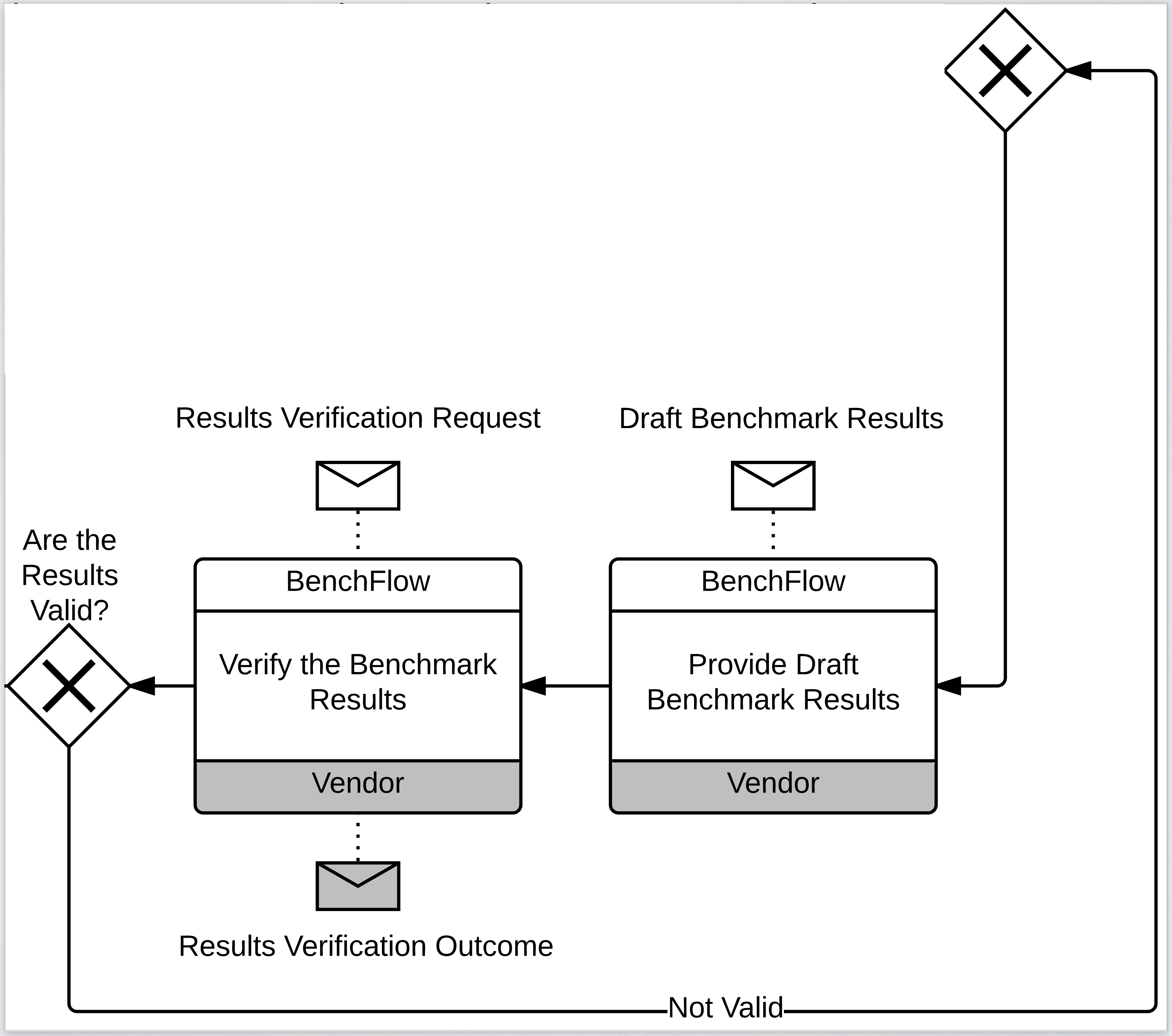
MySQL: Community Server 5.6.26

WfMS A: v7.0.1

Hardware Configuration

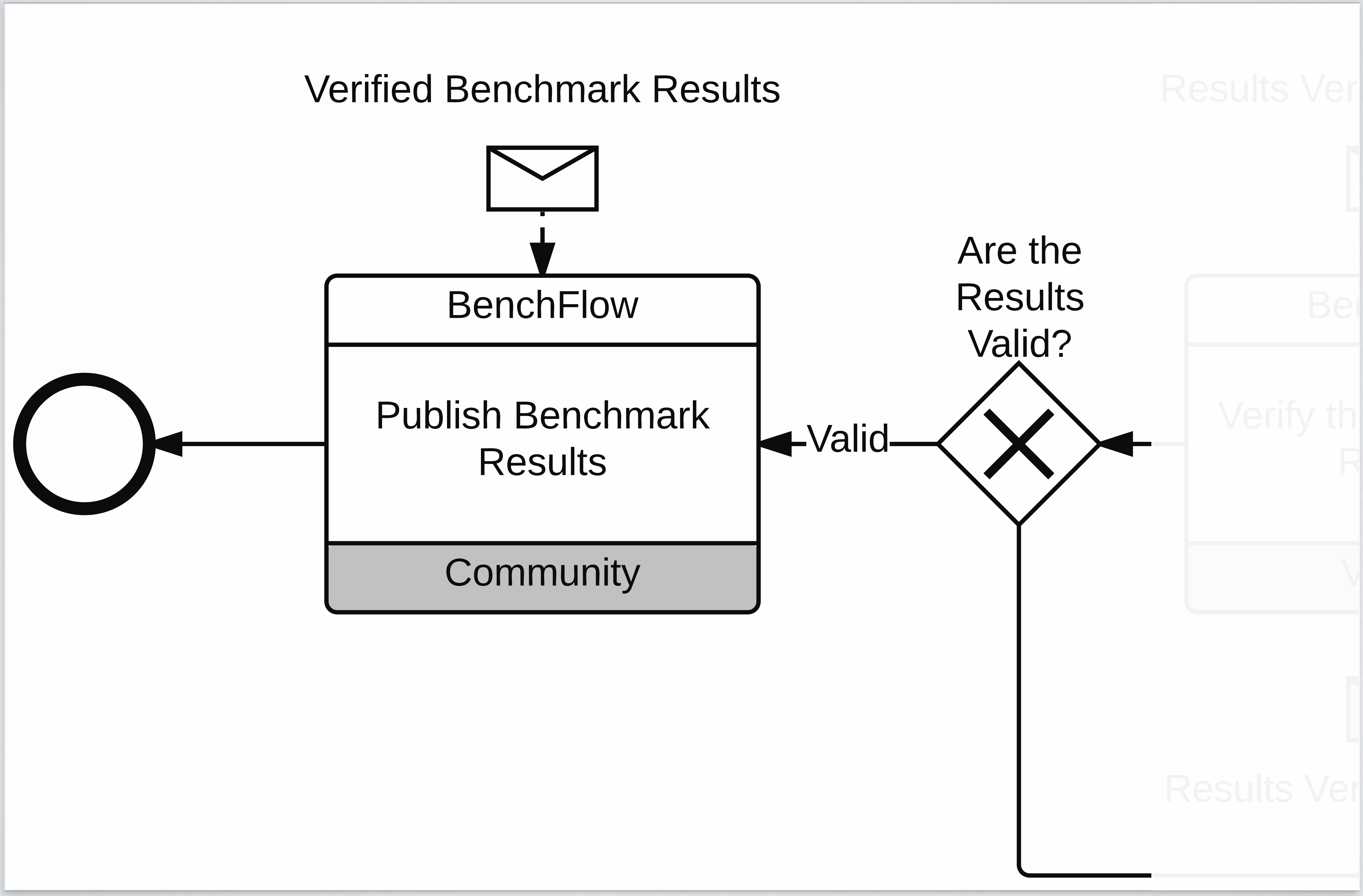
Benchmarking Methodology

executing the benchmark and providing results


Results Verification Request
Draft Benchmark Results
Are the Results Valid?
BenchFlow
Verify the Benchmark Results
Vendor
BenchFlow
Provide Draft Benchmark Results
Vendor
Results Verification Outcome
Not Valid

Benchmarking Methodology

publish benchmark results



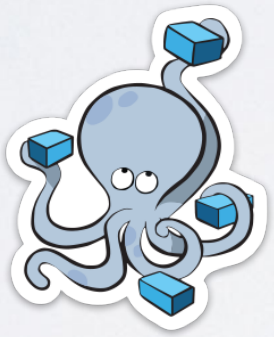
Advantages of using Containers

- Accomplish some Benchmarking Requirement:

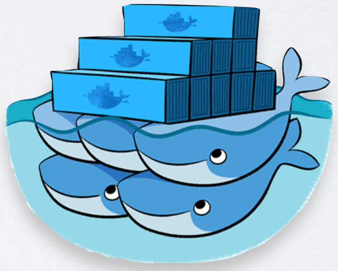
  Portability, Repeatability, Accessibility, Efficiency

Advantages of using Containers

- Accomplish some Benchmarking Requirement:
  Portability, Repeatability, Accessibility, Efficiency
- Common way to deploy systems provided by different vendors



Docker Compose



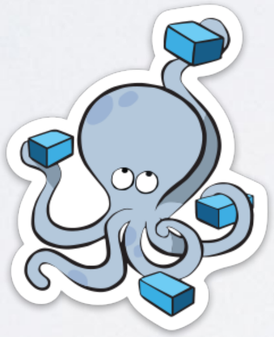
Docker Swarm

Advantages of using Containers

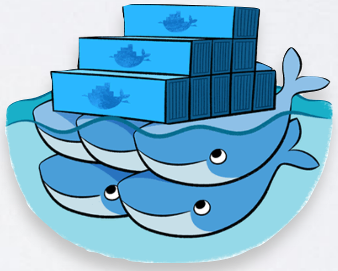
- Accomplish some Benchmarking Requirement:

  Portability, Repeatability, Accessibility, Efficiency

- Common way to deploy systems provided by different vendors



Docker Compose



Docker Swarm

- Standard APIs to access Environment Metrics

First Application of the Methodology

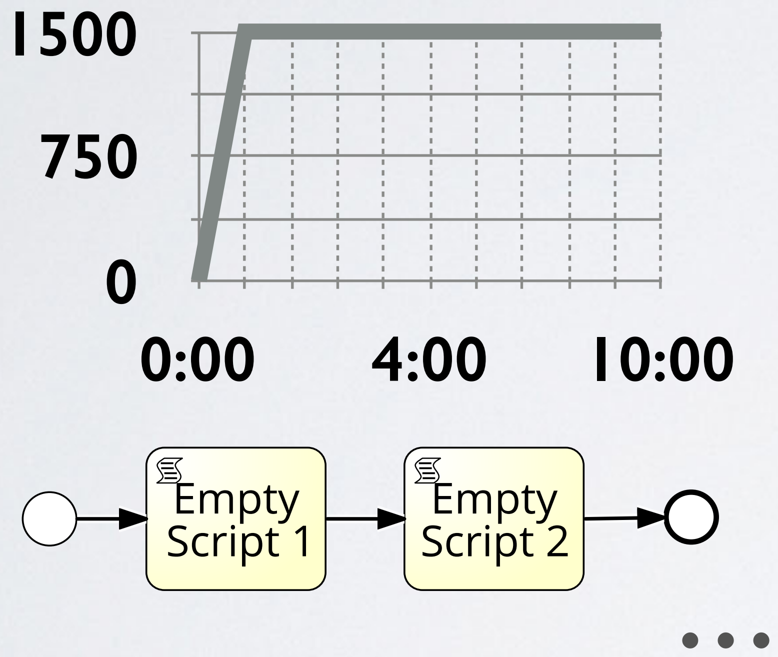
[CAiSE '16]

M. Skouradaki, V. Ferme, C. Pautasso, F. Leymann, A. van Hoorn. **Micro-Benchmarking BPMN 2.0 Workflow Management Systems with Workflow Patterns** . In Proc. of CAiSE '16, June, 2016.

First Application of the Methodology

[CAiSE '16]

M. Skouradaki, V. Ferme, C. Pautasso, F. Leymann, A. van Hoorn. **Micro-Benchmarking BPMN 2.0 Workflow Management Systems with Workflow Patterns** . In Proc. of CAiSE '16, June, 2016.

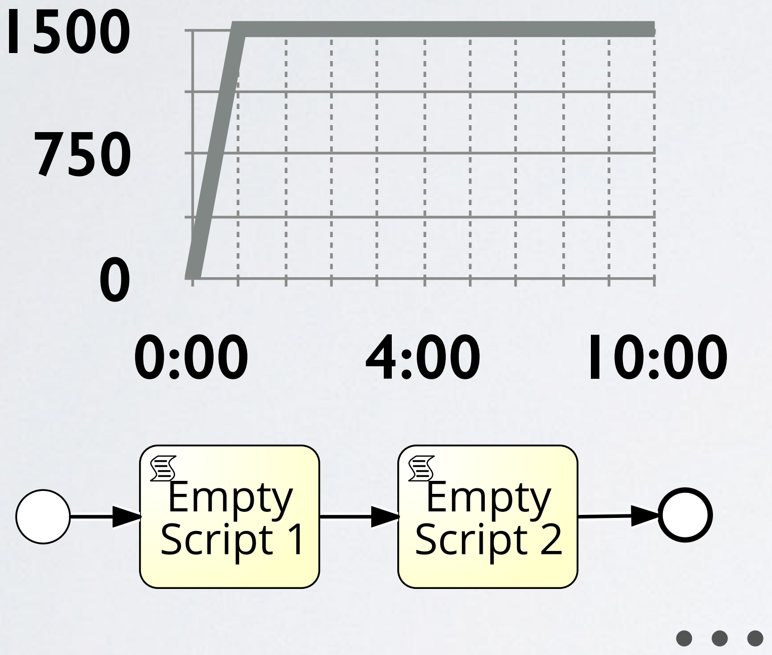


Workload

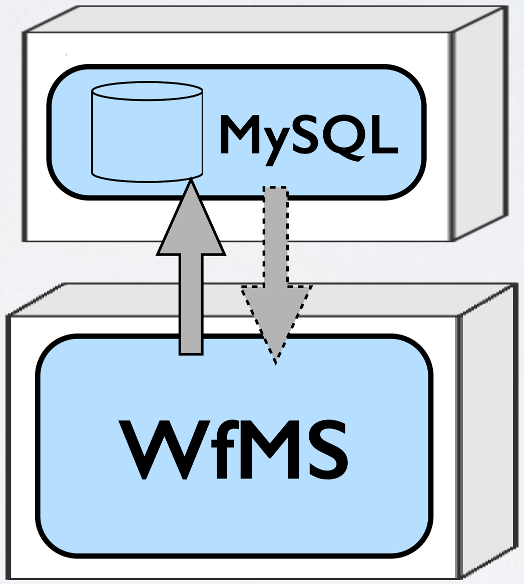
First Application of the Methodology

[CAiSE '16]

M. Skouradaki, V. Ferme, C. Pautasso, F. Leymann, A. van Hoorn. **Micro-Benchmarking BPMN 2.0 Workflow Management Systems with Workflow Patterns** . In Proc. of CAiSE '16, June, 2016.



Workload

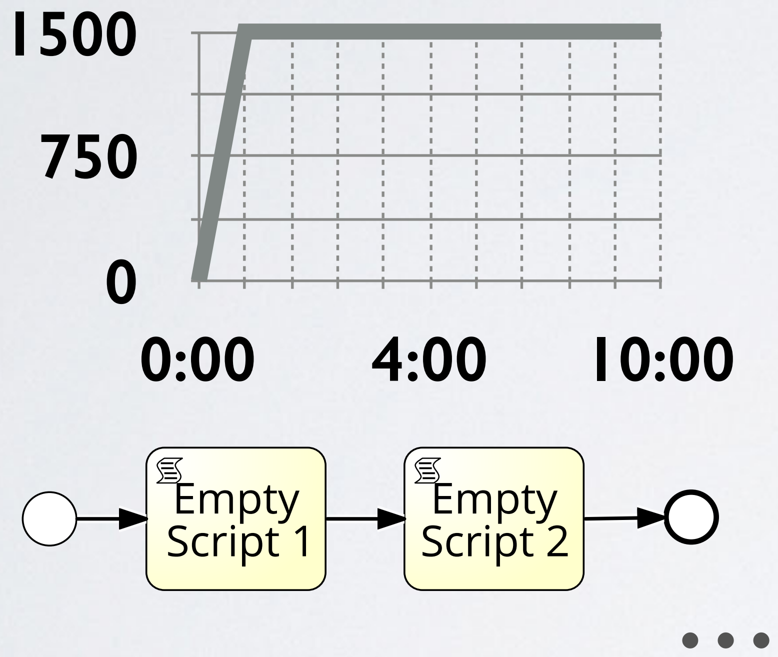


3 WfMSs

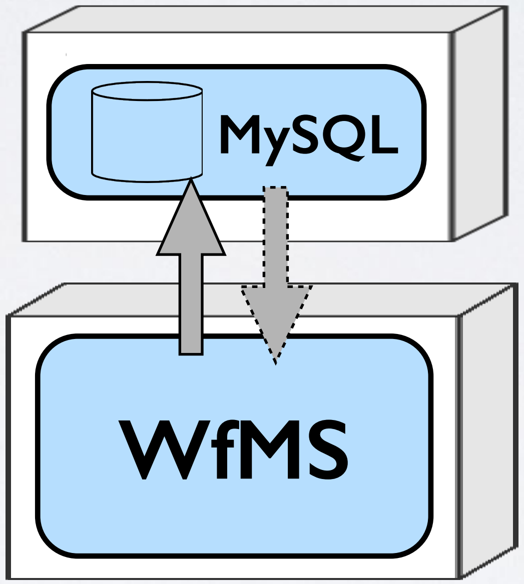
First Application of the Methodology

[CAiSE '16]

M. Skouradaki, V. Ferme, C. Pautasso, F. Leymann, A. van Hoorn. **Micro-Benchmarking BPMN 2.0 Workflow Management Systems with Workflow Patterns** . In Proc. of CAiSE '16, June, 2016.



Workload



3 WfMSs

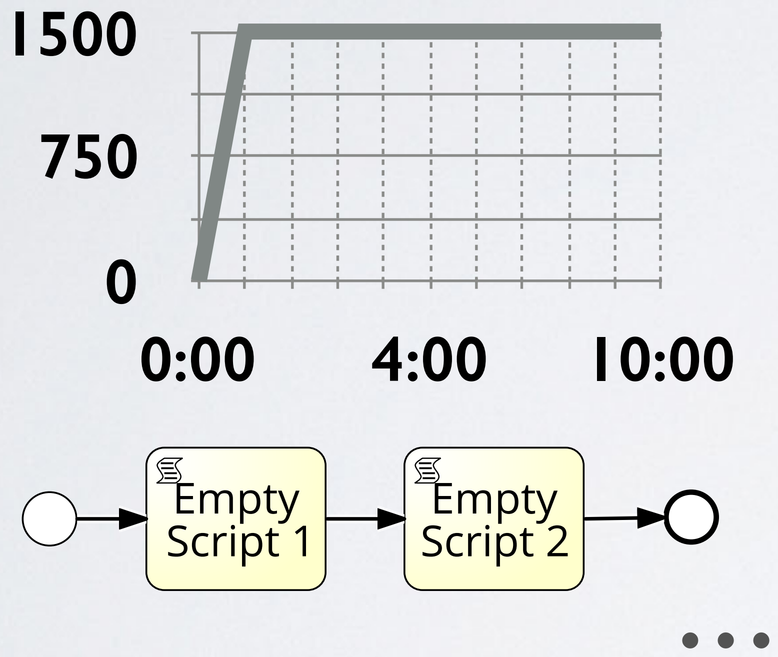
- Engine Level
- Process Level
- Environment

Metrics

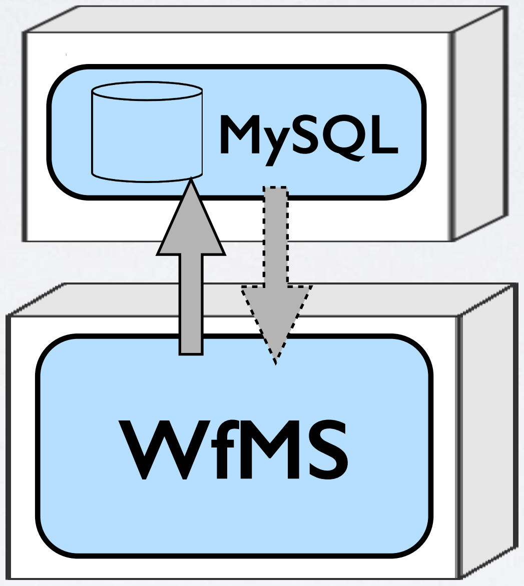
First Application of the Methodology

[CAiSE '16]

M. Skouradaki, V. Ferme, C. Pautasso, F. Leymann, A. van Hoorn. **Micro-Benchmarking BPMN 2.0 Workflow Management Systems with Workflow Patterns** . In Proc. of CAiSE '16, June, 2016.



Workload



3 WfMSs

- Engine Level
- Process Level
- Environment

Metrics

Results: relevant differences among WfMSs

Future Work

- Continue to Apply and Improve the Methodology

Future Work

- Continue to Apply and Improve the Methodology
- Involve more Vendors and Researchers as part of the Benchmarking Effort

Future Work

- Continue to Apply and Improve the Methodology
- Involve more **Vendors** and Researchers as part of the Benchmarking Effort

Future Work

- Continue to Apply and Improve the Methodology
- Involve more **Vendors** and Researchers as part of the Benchmarking Effort

Activiti™

camunda

jBPM

Future Work

- Continue to Apply and Improve the Methodology
- Involve more **Vendors and Researchers** as part of the Benchmarking Effort

Future Work

- Continue to Apply and Improve the Methodology
- Involve more **Vendors and Researchers** as part of the Benchmarking Effort

1st International Workshop on Performance and Conformance of Workflow Engines

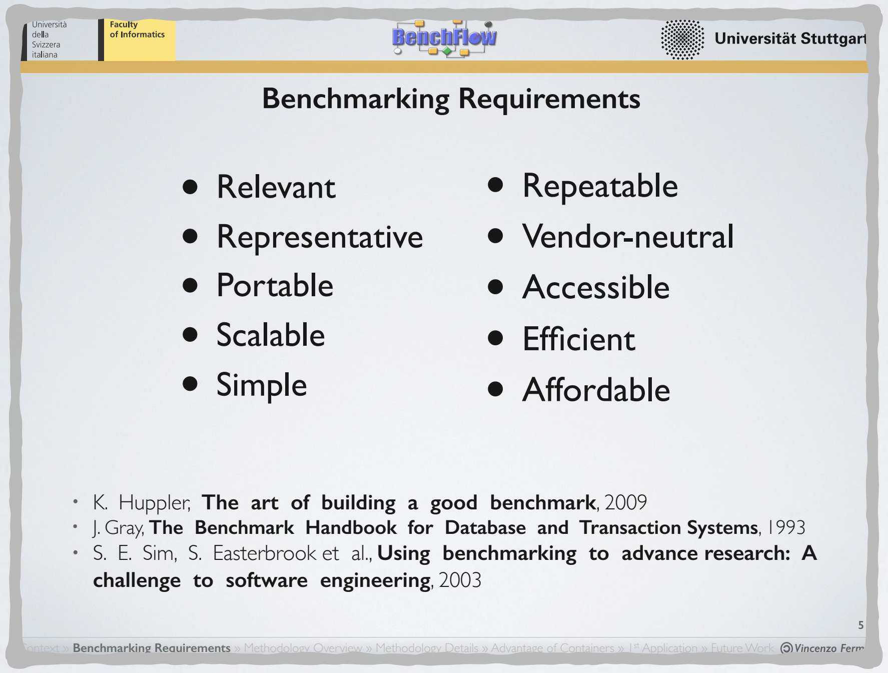
Workshop
September 5th, 2016

ESOCC 2016
Vienna, Austria

http://uniba-dsg.github.io/peace-ws/

Highlights

Highlights



Università della Svizzera italiana

Faculty of Informatics

Universität Stuttgart

Benchmarking Requirements

- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

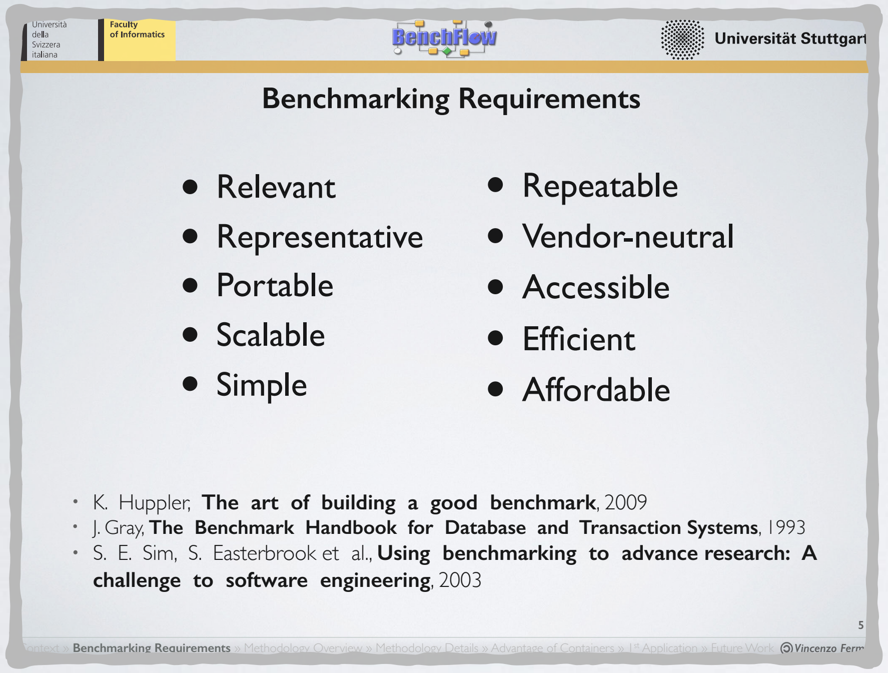
- K. Huppler, **The art of building a good benchmark**, 2009
- J. Gray, **The Benchmark Handbook for Database and Transaction Systems**, 1993
- S. E. Sim, S. Easterbrook et al., **Using benchmarking to advance research: A challenge to software engineering**, 2003

5

Context » Benchmarking Requirements » Methodology Overview » Methodology Details » Advantage of Containers » 1st Application » Future Work Vincenzo Ferme

Benchmarking Requirements

Highlights



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

Benchmarking Requirements

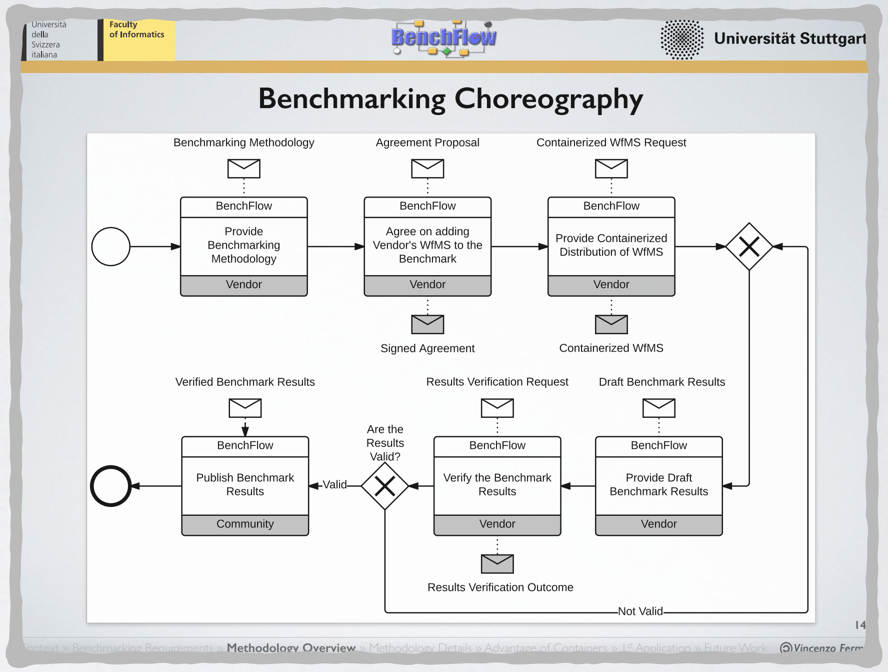
- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

- K. Huppler, **The art of building a good benchmark**, 2009
- J. Gray, **The Benchmark Handbook for Database and Transaction Systems**, 1993
- S. E. Sim, S. Easterbrook et al., **Using benchmarking to advance research: A challenge to software engineering**, 2003

5

Context » **Benchmarking Requirements** » Methodology Overview » Methodology Details » Advantage of Containers » 1st Application » Future Work Ⓒ Vincenzo Ferm

Benchmarking Requirements



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

Benchmarking Choreography

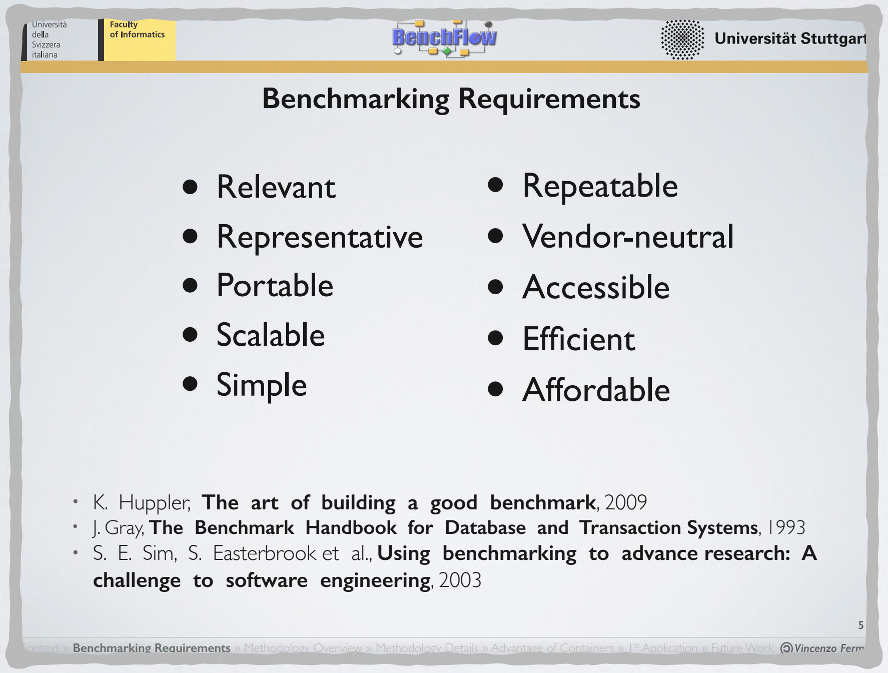
Benchmarking Methodology
BenchFlow
Provide Benchmarking Methodology
Vendor
Agreement Proposal
BenchFlow
Agree on adding Vendor's WfMS to the Benchmark
Vendor
Signed Agreement
Containerized WfMS Request
BenchFlow
Provide Containerized Distribution of WfMS
Vendor
Containerized WfMS
Verified Benchmark Results
BenchFlow
Publish Benchmark Results
Community
Valid
Are the Results Valid?
Results Verification Request
BenchFlow
Verify the Benchmark Results
Vendor
Results Verification Outcome
Draft Benchmark Results
BenchFlow
Provide Draft Benchmark Results
Vendor
Not Valid

14

Context » Benchmarking Requirements » **Methodology Overview** » Methodology Details » Advantage of Containers » 1st Application » Future Work Ⓒ Vincenzo Ferm

Benchmarking Methodology

Highlights



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

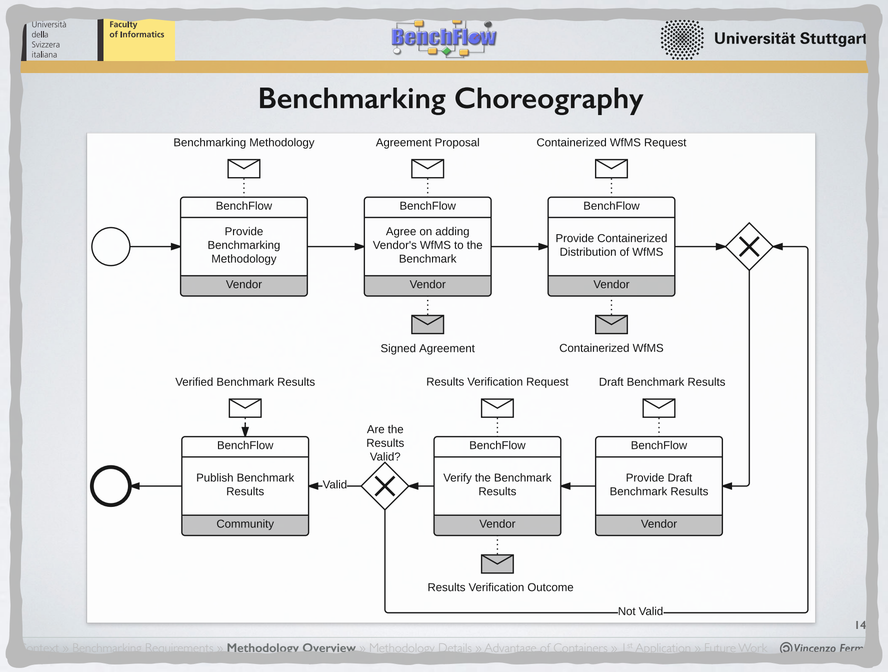
Benchmarking Requirements

- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

- K. Huppler, **The art of building a good benchmark**, 2009
- J. Gray, **The Benchmark Handbook for Database and Transaction Systems**, 1993
- S. E. Sim, S. Easterbrook et al., **Using benchmarking to advance research: A challenge to software engineering**, 2003

5

Benchmarking Requirements



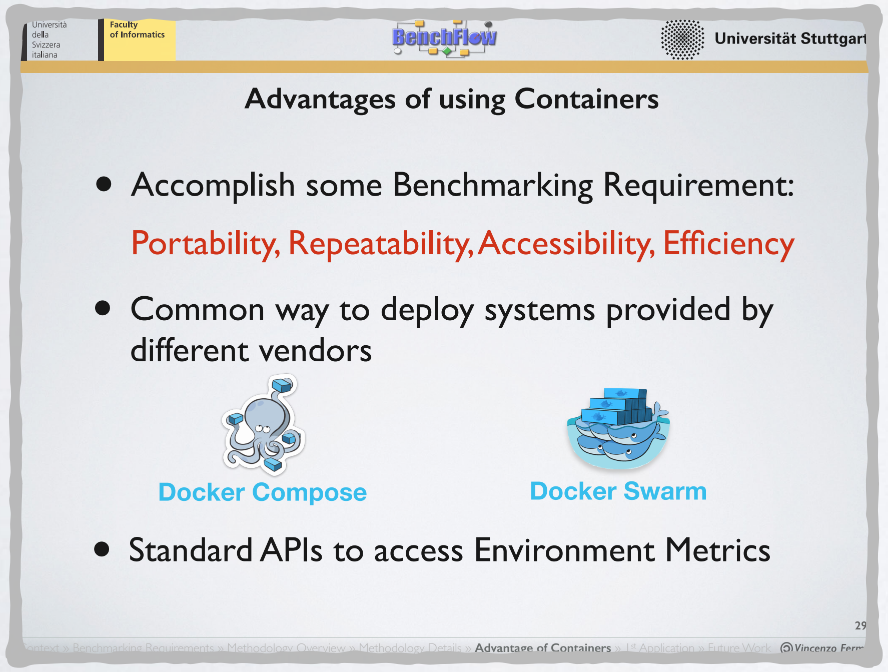
Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

Benchmarking Choreography

Benchmarking Methodology
Agreement Proposal
Containerized WfMS Request
BenchFlow
Provide Benchmarking Methodology
Vendor
BenchFlow
Agree on adding Vendor's WfMS to the Benchmark
Vendor
BenchFlow
Provide Containerized Distribution of WfMS
Vendor
Signed Agreement
Containerized WfMS
Verified Benchmark Results
Results Verification Request
Draft Benchmark Results
BenchFlow
Publish Benchmark Results
Community
Are the Results Valid?
Valid
BenchFlow
Verify the Benchmark Results
Vendor
BenchFlow
Provide Draft Benchmark Results
Vendor
Results Verification Outcome
Not Valid

14

Benchmarking Methodology



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

Advantages of using Containers

- Accomplish some Benchmarking Requirement: Portability, Repeatability, Accessibility, Efficiency
- Common way to deploy systems provided by different vendors

Docker Compose

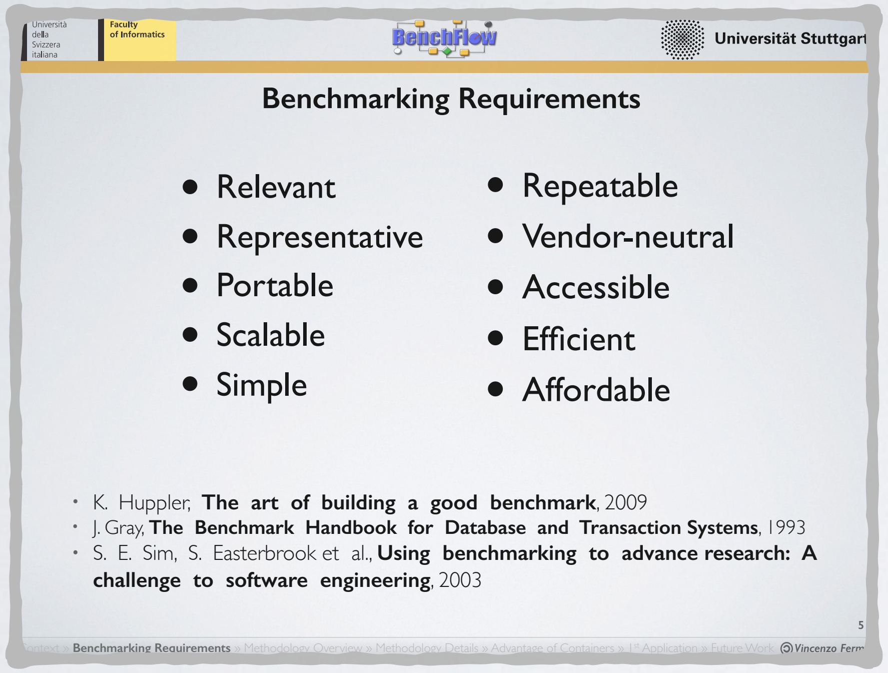
Docker Swarm

- Standard APIs to access Environment Metrics

29

Advantages of Containers

Highlights



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

Benchmarking Requirements

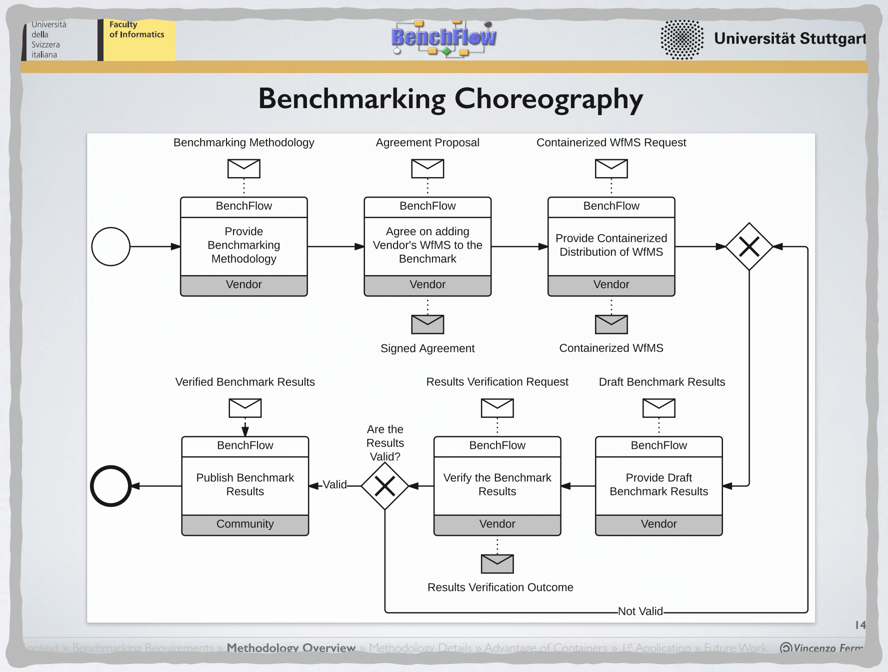
- Relevant
- Representative
- Portable
- Scalable
- Simple
- Repeatable
- Vendor-neutral
- Accessible
- Efficient
- Affordable

- K. Huppler, **The art of building a good benchmark**, 2009
- J. Gray, **The Benchmark Handbook for Database and Transaction Systems**, 1993
- S. E. Sim, S. Easterbrook et al., **Using benchmarking to advance research: A challenge to software engineering**, 2003

5

Context » **Benchmarking Requirements** » Methodology Overview » Methodology Details » Advantage of Containers » 1st Application » Future Work Vincenzo Ferm

Benchmarking Requirements



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

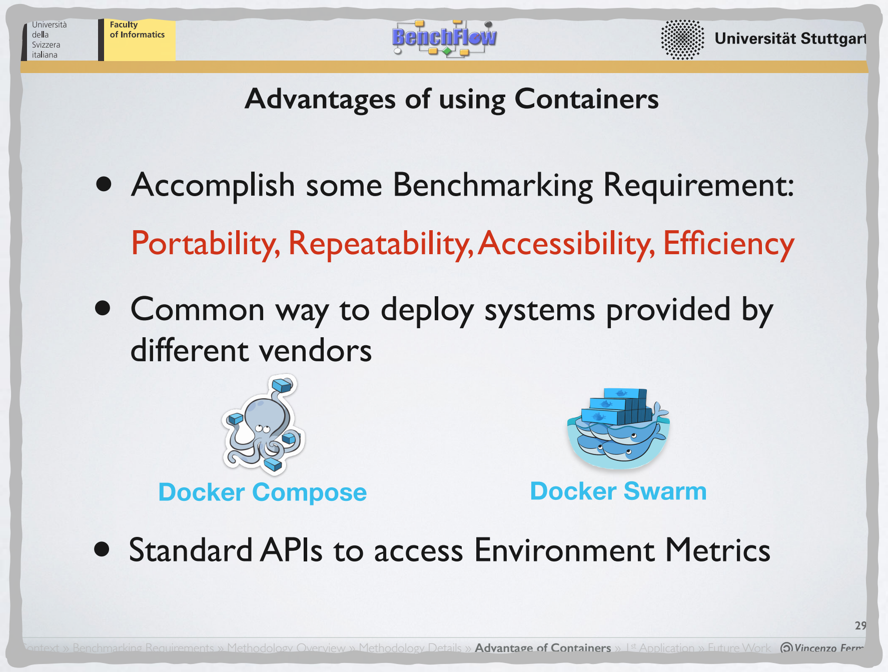
Benchmarking Choreography

Benchmarking Methodology
BenchFlow
Provide Benchmarking Methodology
Vendor
Agreement Proposal
BenchFlow
Agree on adding Vendor's WfMS to the Benchmark
Vendor
Signed Agreement
Containerized WfMS Request
BenchFlow
Provide Containerized Distribution of WfMS
Vendor
Containerized WfMS
Verified Benchmark Results
BenchFlow
Publish Benchmark Results
Community
Valid
Are the Results Valid?
Results Verification Request
BenchFlow
Verify the Benchmark Results
Vendor
Results Verification Outcome
Draft Benchmark Results
BenchFlow
Provide Draft Benchmark Results
Vendor
Not Valid

14

Context » Benchmarking Requirements » **Methodology Overview** » Methodology Details » Advantage of Containers » 1st Application » Future Work Vincenzo Ferm

Benchmarking Methodology



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

Advantages of using Containers

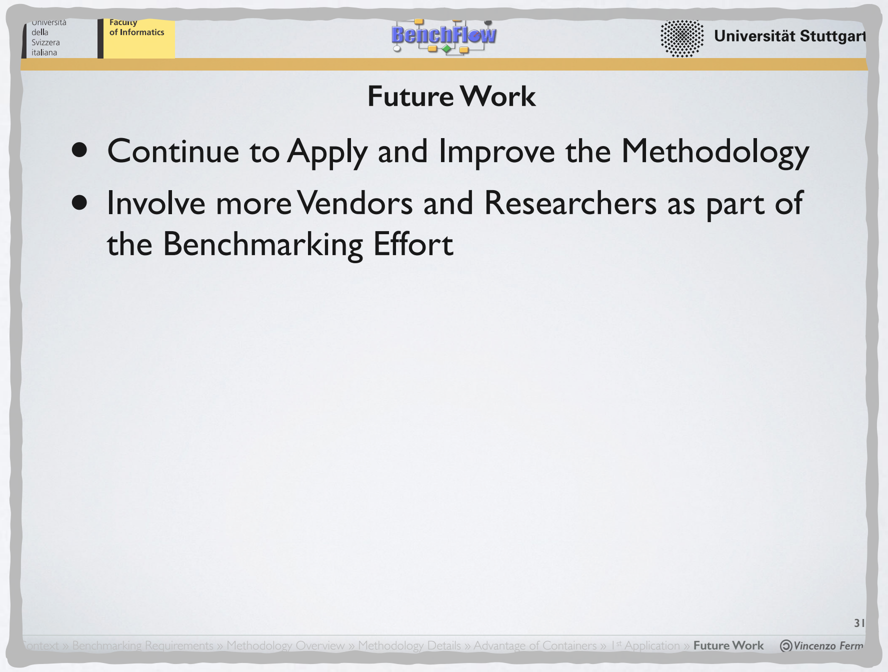
- Accomplish some Benchmarking Requirement:
  Portability, Repeatability, Accessibility, Efficiency
- Common way to deploy systems provided by different vendors

- Standard APIs to access Environment Metrics

29

Context » Benchmarking Requirements » Methodology Overview » Methodology Details » **Advantage of Containers** » 1st Application » Future Work Vincenzo Ferm

Advantages of Containers



Università della Svizzera italiana | Faculty of Informatics | BenchFlow | Universität Stuttgart

Future Work

- Continue to Apply and Improve the Methodology
- Involve more Vendors and Researchers as part of the Benchmarking Effort

31

Context » Benchmarking Requirements » Methodology Overview » Methodology Details » Advantage of Containers » 1st Application » **Future Work** Vincenzo Ferm

Future Work

A CONTAINER-CENTRIC METHODOLOGY FOR BENCHMARKING WORKFLOW MANAGEMENT SYSTEMS

benchflow

benchflow

http://benchflow.inf.usi.ch

✉ vincenzo.ferme@usi.ch

Vincenzo Ferme (@VincenzoFerme), Ana Ivanchikj, Prof. Cesare Pautasso
Faculty of Informatics
University of Lugano (USI), Switzerland

Marigianna Skouradaki, Prof. Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart, Germany

BACKUP SLIDES

Vincenzo Ferme, Ana Ivanchikj,
Prof. Cesare Pautasso
Faculty of Informatics
University of Lugano (USI), Switzerland

Marigianna Skouradaki,
Prof. Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart, Germany

Published Work

[SSP '14]
M. Skouradaki, D. H. Roller, F. Leymann, V. Ferme, and C. Pautasso. **Technical open challenges on benchmarking workflow management systems**. In Proc. of the 2014 Symposium on Software Performance, SSP 2014, pages 105–112, 2014.

[BTW '15]
C. Pautasso, V. Ferme, D. Roller, F. Leymann, and M. Skouradaki. **Towards workflow benchmarking: Open research challenges**. In Proc. of the 16th conference on Database Systems for Business, Technology, and Web, BTW 2015, pages 331–350, 2015.

[ICPE '15]
M. Skouradaki, D. H. Roller, L. Frank, V. Ferme, and C. Pautasso. **On the Road to Benchmarking BPMN 2.0 Workflow Engines**. In Proc. of the 6th ACM/SPEC International Conference on Performance Engineering, ICPE '15, pages 301–304, 2015.

Published Work

[CLOSER '15]

M. Skouradaki, V. Ferme, F. Leymann, C. Pautasso, and D. H. Roller. **"BPELanon": Protect business processes on the cloud.** In Proc. of the 5th International Conference on Cloud Computing and Service Science, CLOSER 2015. SciTePress, 2015.

[SOSE '15]

M. Skouradaki, K. Goerlach, M. Hahn, and F. Leymann. **Application of Sub-Graph Isomorphism to Extract Reoccurring Structures from BPMN 2.0 Process Models**. In Proc. of the 9th International IEEE Symposium on Service-Oriented System Engineering, SOSE 2015, 2015.

[BPM '15]

V. Ferme, A. Ivanchikj, C. Pautasso. **A Framework for Benchmarking BPMN 2.0 Workflow Management Systems**. In Proc. of the 13th International Conference on Business Process Management, BPM '15, pages 251-259, 2015.

Published Work

[BPMD '15]
A. Ivanchikj, V. Ferme, C. Pautasso. **BPMeter: Web Service and Application for Static Analysis of BPMN 2.0 Collections**. In Proc. of the 13th International Conference on Business Process Management [Demo], BPM '15, pages 30-34, 2015.

[ICPE '16]
V. Ferme, and C. Pautasso. **Integrating Faban with Docker for Performance Benchmarking**. In Proc. of the 7th ACM/SPEC International Conference on Performance Engineering, ICPE '16, 2016.

[CAiSE '16]
M. Skouradaki, V. Ferme, C. Pautasso, F. Leymann, A. van Hoorn. **Micro-Benchmarking BPMN 2.0 Workflow Management Systems with Workflow Patterns** . In Proc. of the 28th International Conference on Advanced Information Systems Engineering, CAiSE '16, 2016.

Docker Performance

[IBM '14]

W. Felter, A. Ferreira, R. Rajamony, and J. Rubio. **An updated performance comparison of virtual machines and Linux containers.** IBM Research Report, 2014.

> "Our results show that **containers result in equal or better performance than VMs in almost all cases.**"

> "Although **containers themselves have almost no overhead**, **Docker is not without performance gotchas**. Docker volumes have noticeably better performance than files stored in AUFS. Docker's NAT also introduces overhead for workloads with high packet rates. These **features** represent a **tradeoff between ease of management and performance and should be considered on a case-by-case basis.**"

BenchFlow Configures Docker for Performance by Default