WORKFLOW ENGINE PERFORMANCE BENCHMARKING WITH BENCHFLOW

Vincenzo Ferme
Faculty of Informatics
USI Lugano, Switzerland

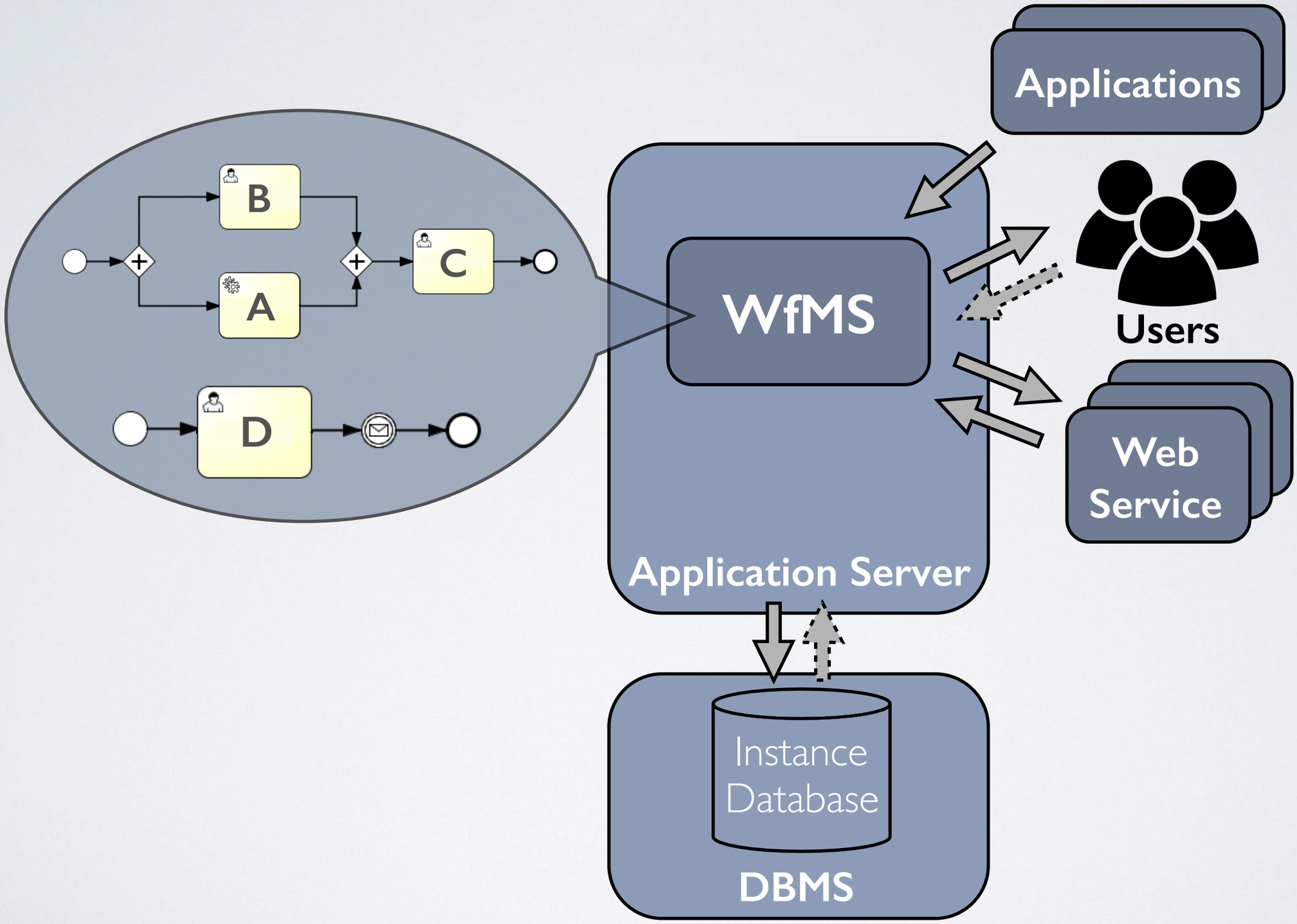
ESOCC 2016

PEACE
Performance And Conformance Benchmarking Of Workflow Engines

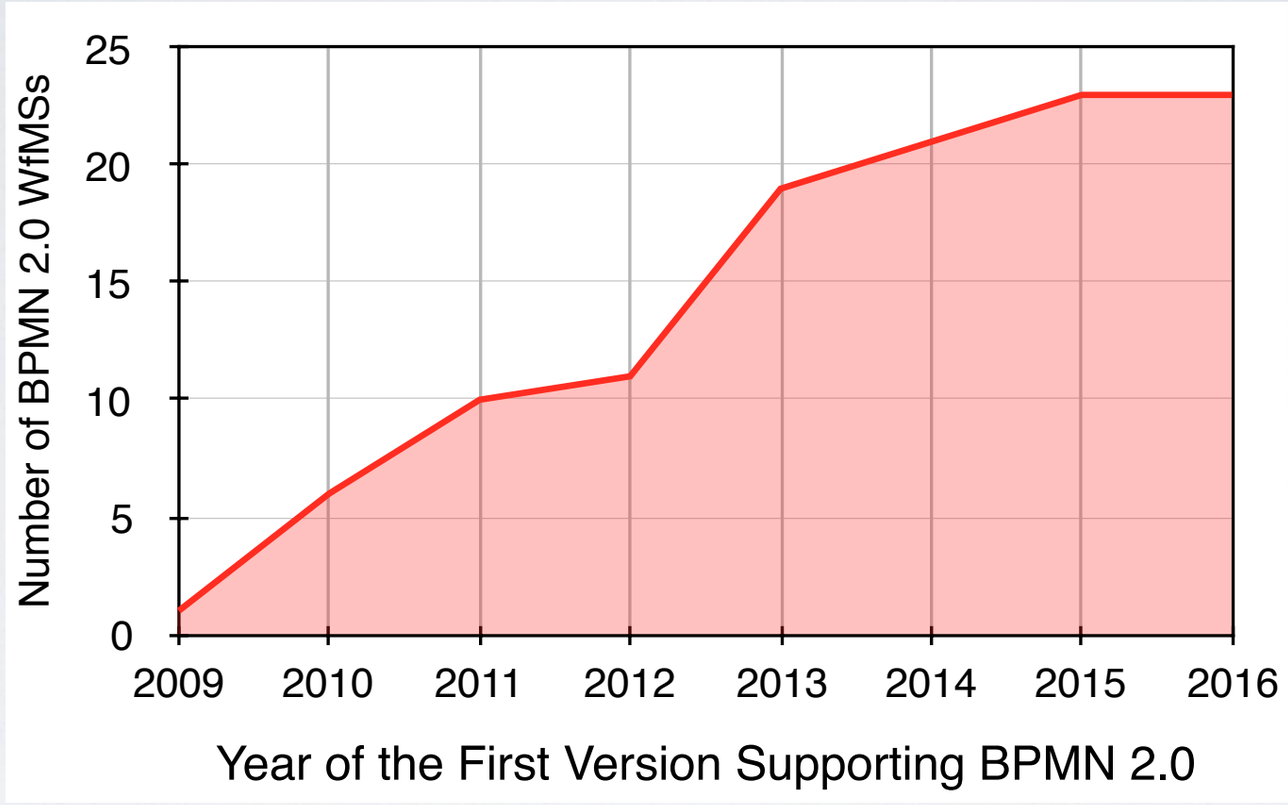
The BenchFlow Project

“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.”

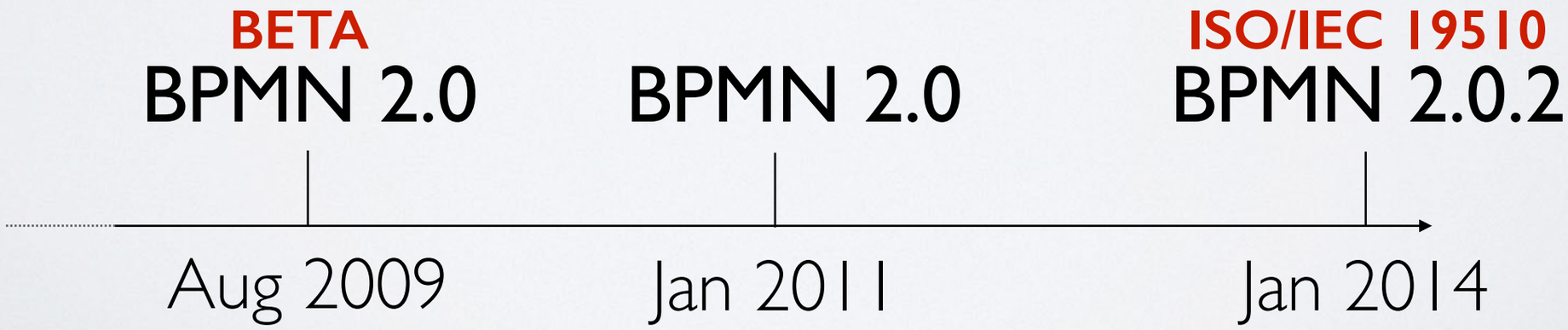
What is a Workflow Management System?



Many Vendors of BPMN 2.0 WfMSs



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

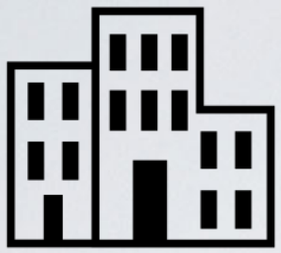


Why do We Need a Benchmark?

end-users, vendors, developers

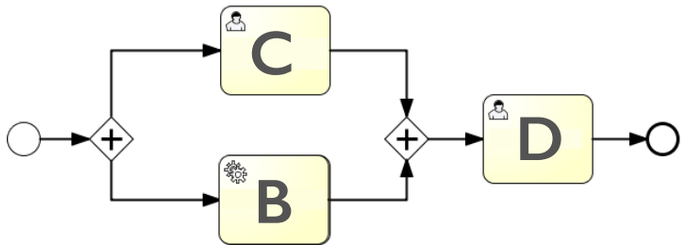
Why do We Need a Benchmark?

end-users, vendors, developers



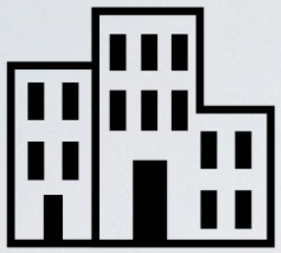
1. How to choose the best WfMS according to the company's technical requirements?

2. How to choose the best WfMS according to the company's business process models (workflows)?

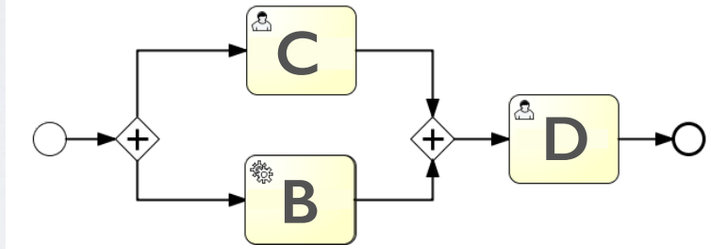


Why do We Need a Benchmark?

end-users, vendors, developers



1. How to choose the best WfMS according to the company's technical requirements?



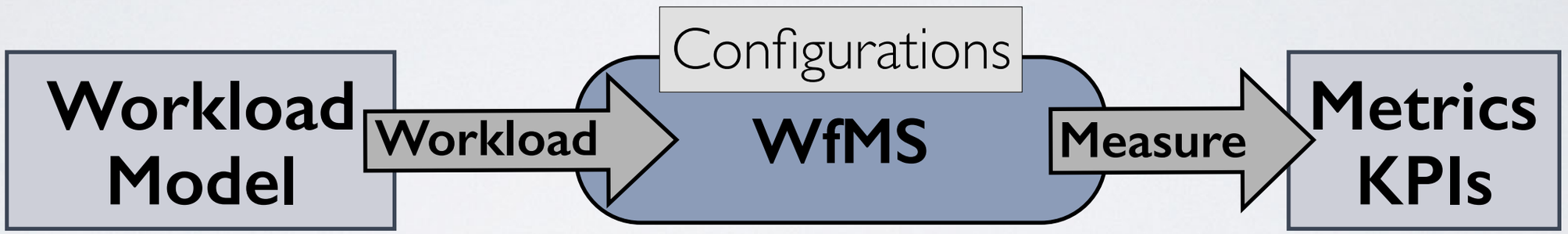
2. How to choose the best WfMS according to the company's business process models (workflows)?

3. How to evaluate performance improvements during WfMS's development?

4. How to identify WfMS's bottlenecks?

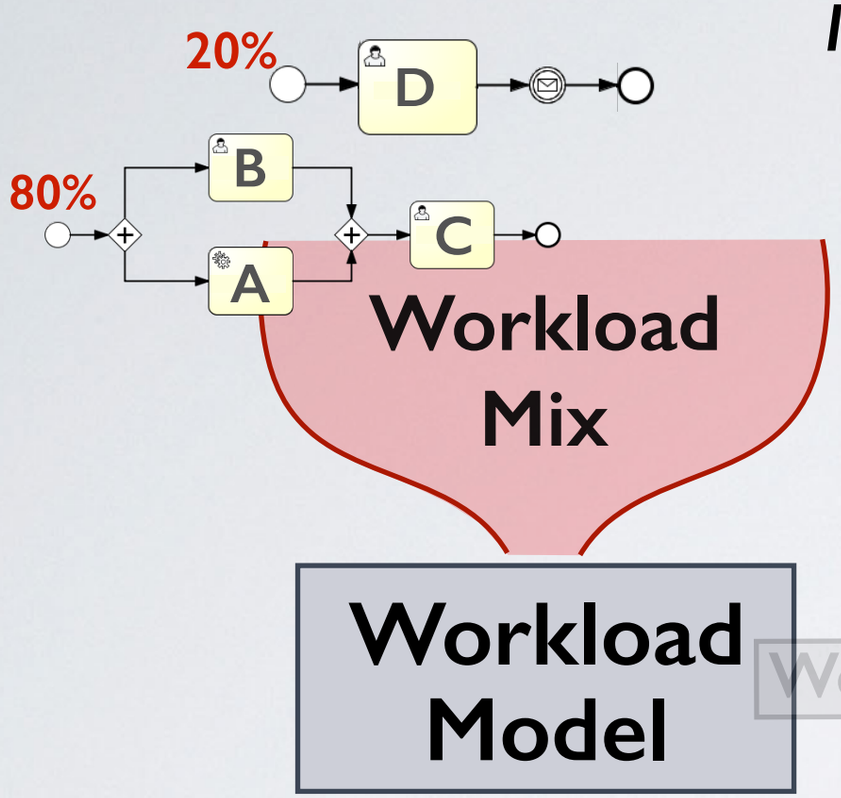
The BenchFlow Benchmarking Process

Input/Process/Output Model



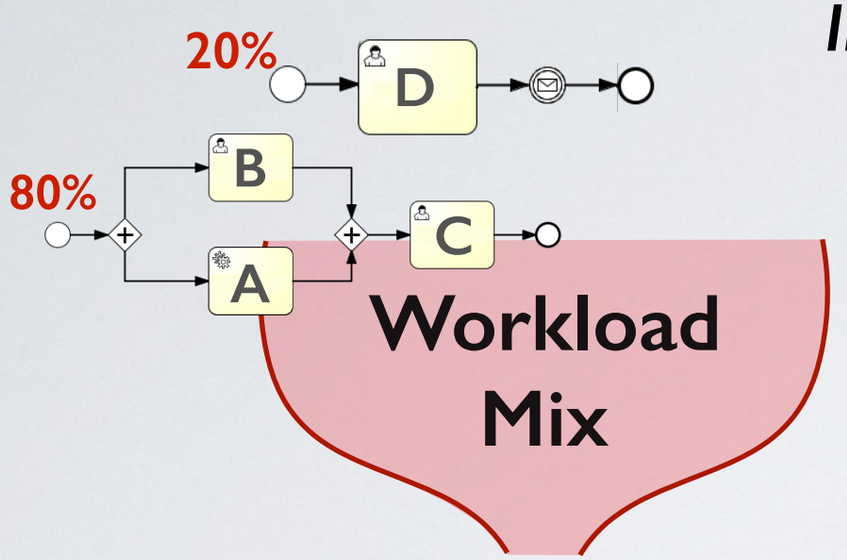
The BenchFlow Benchmarking Process

Input/Process/Output Model



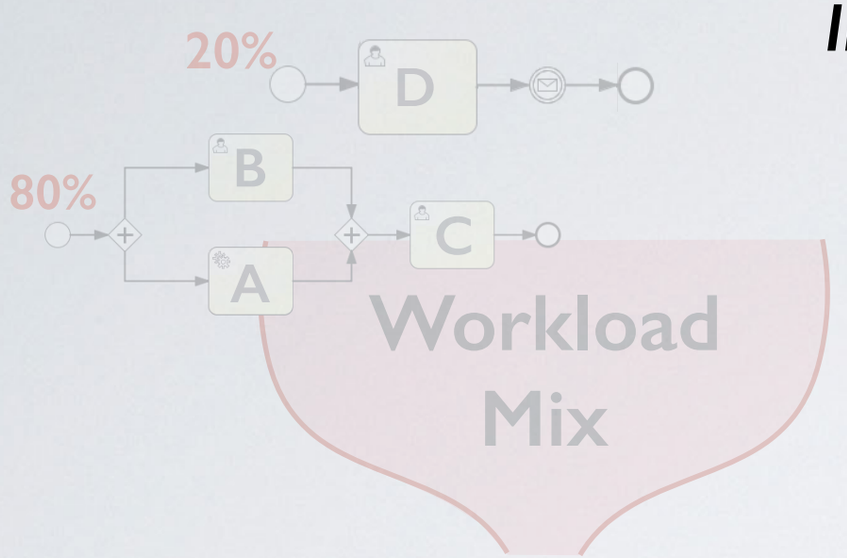
The BenchFlow Benchmarking Process

Input/Process/Output Model



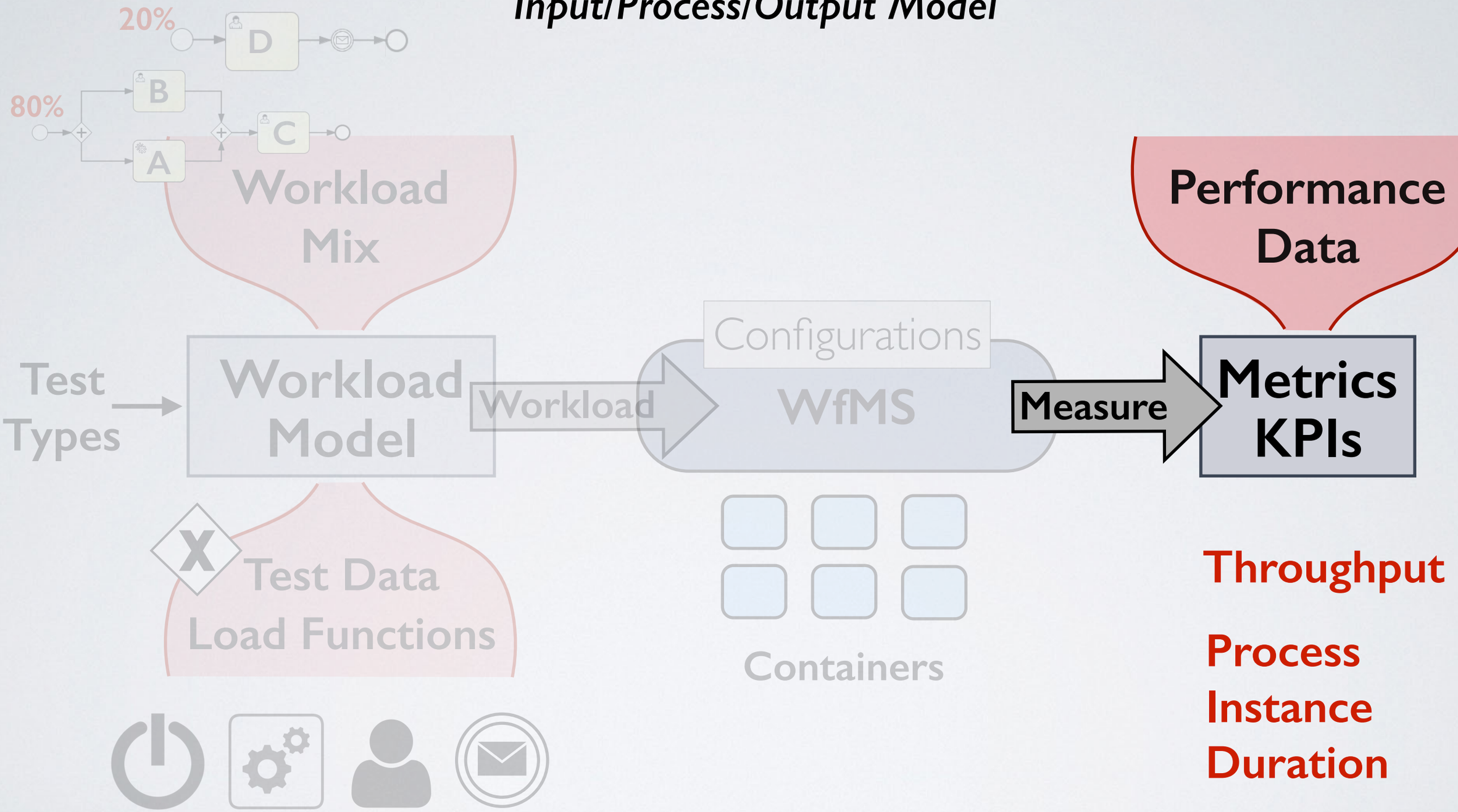
The BenchFlow Benchmarking Process

Input/Process/Output Model

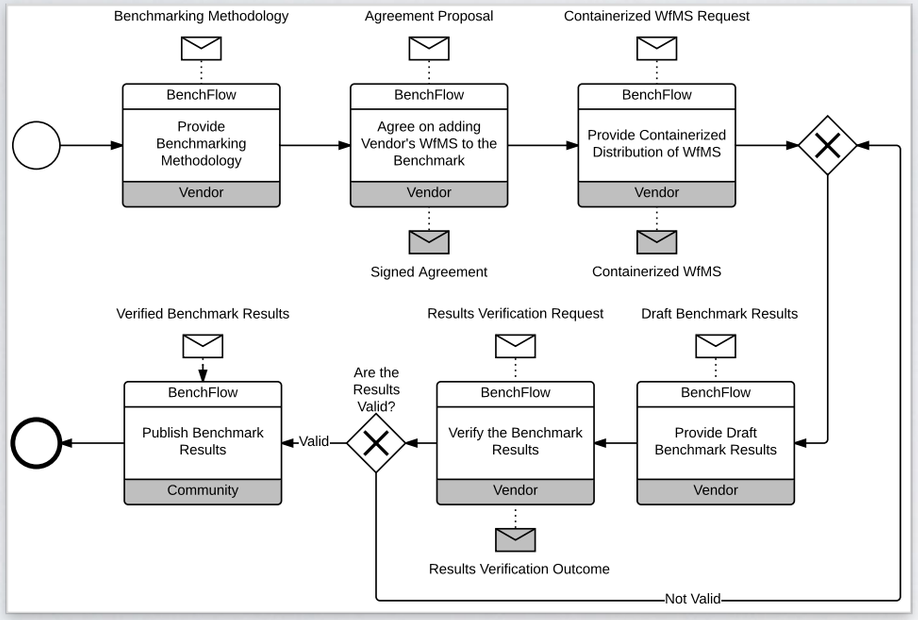


The BenchFlow Benchmarking Process

Input/Process/Output Model

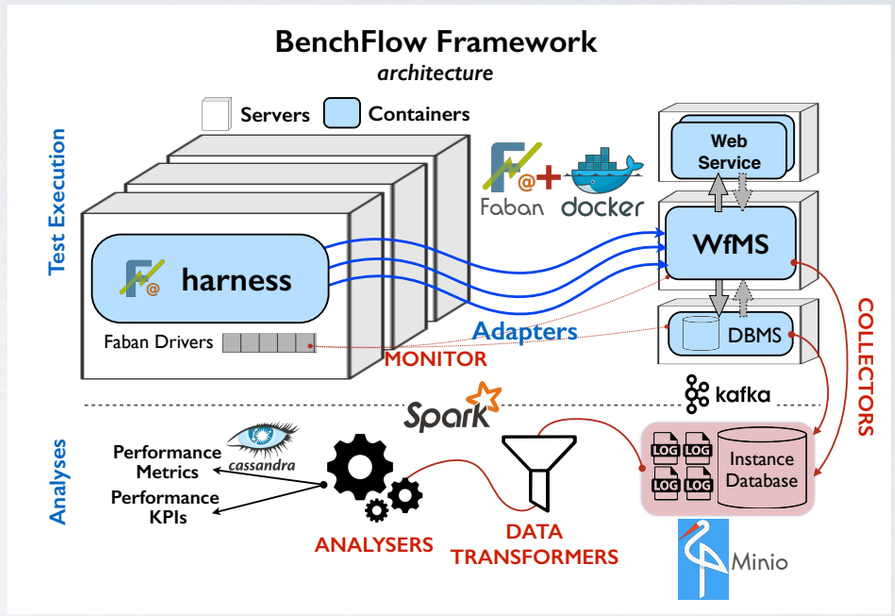


Container Based Methodology and Framework



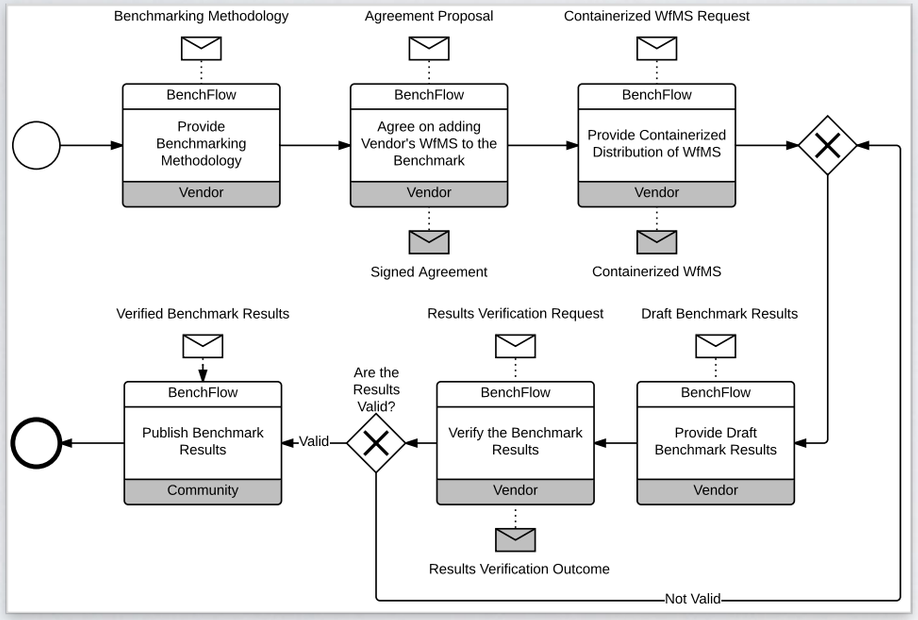
Methodology

+



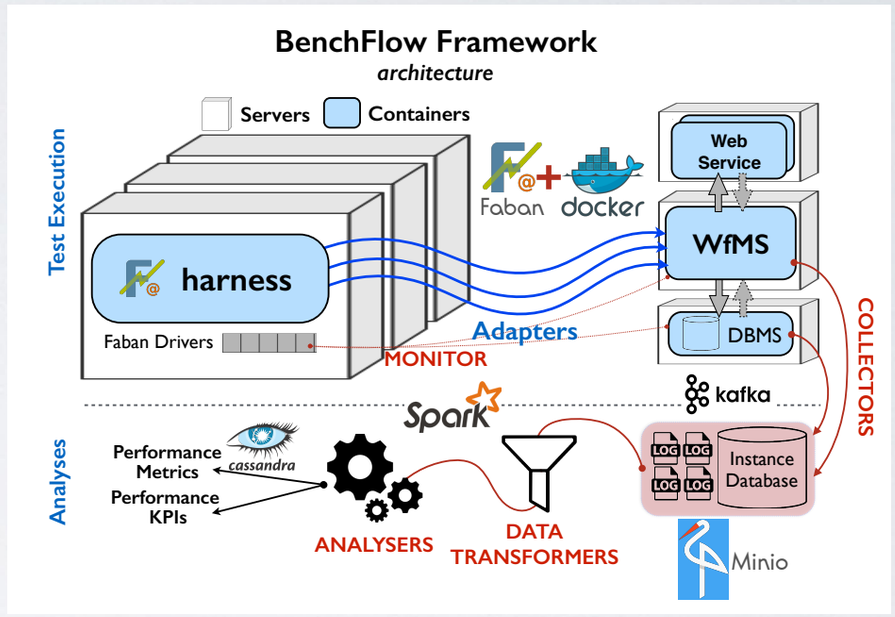
Framework

Container Based Methodology and Framework



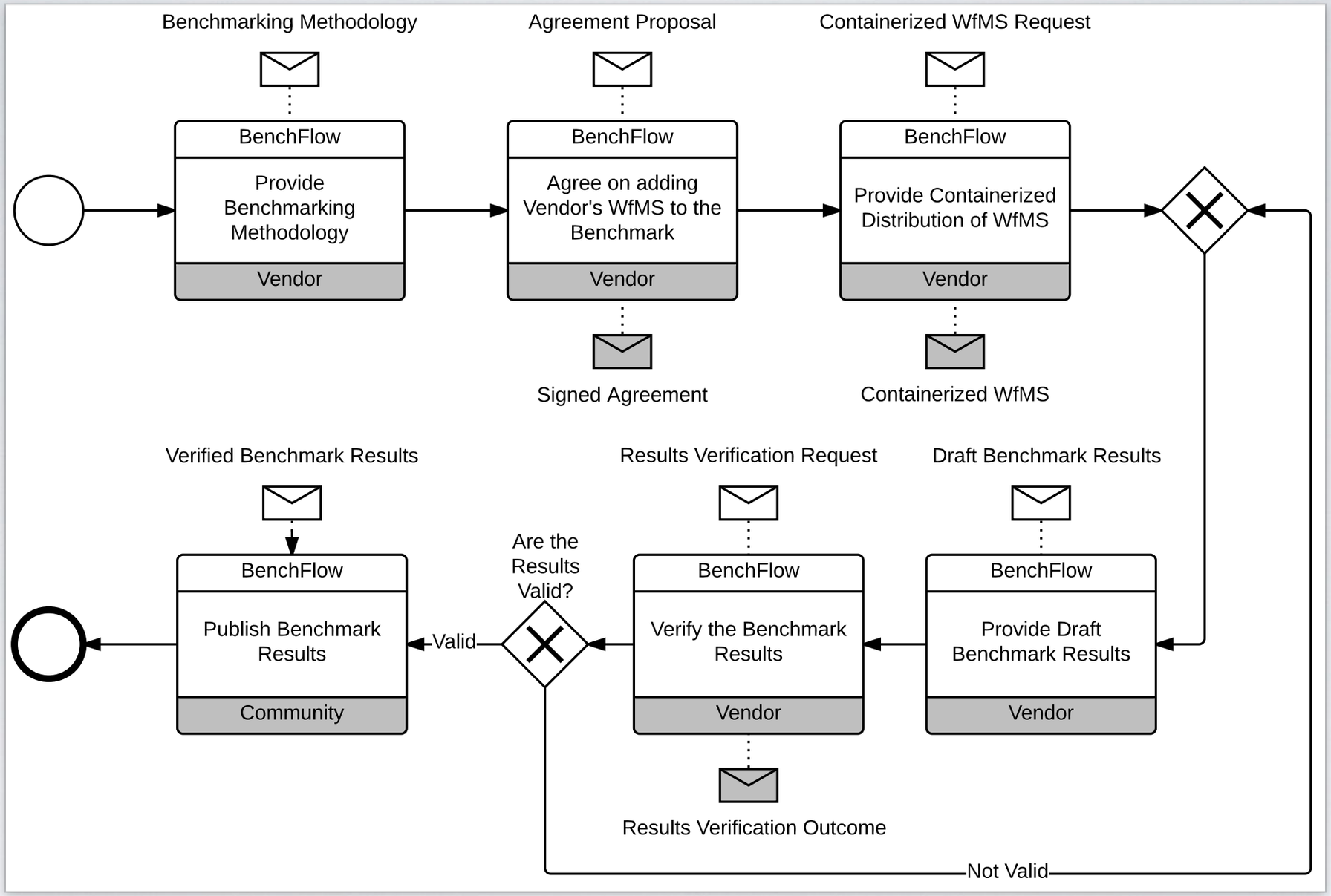
Methodology

+



Framework

Provides **Great Advantages** for **Automation** and **Reproducibility** of Results, while Ensuring **Negligible Performance Impact**



[CLOSER '16]

Container-centric Methodology for Benchmarking Workflow Management Systems.

BenchFlow Benchmarking Methodology

requirements from the WfMS

Availability of APIs

- Deploy Process
- Start Process Instance
- Users API
- Web Service APIs
- Events APIs

BenchFlow Benchmarking Methodology

requirements from the WfMS

Availability of APIs

- Deploy Process
- Start Process Instance
- Users API
- Web Service APIs
- Events APIs

Availability of Timing Data

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

BenchFlow Benchmarking Methodology

requirements from the WfMS

Availability of APIs

- Deploy Process
- Start Process Instance
- Users API
- Web Service APIs
- Events APIs

Availability of Timing Data

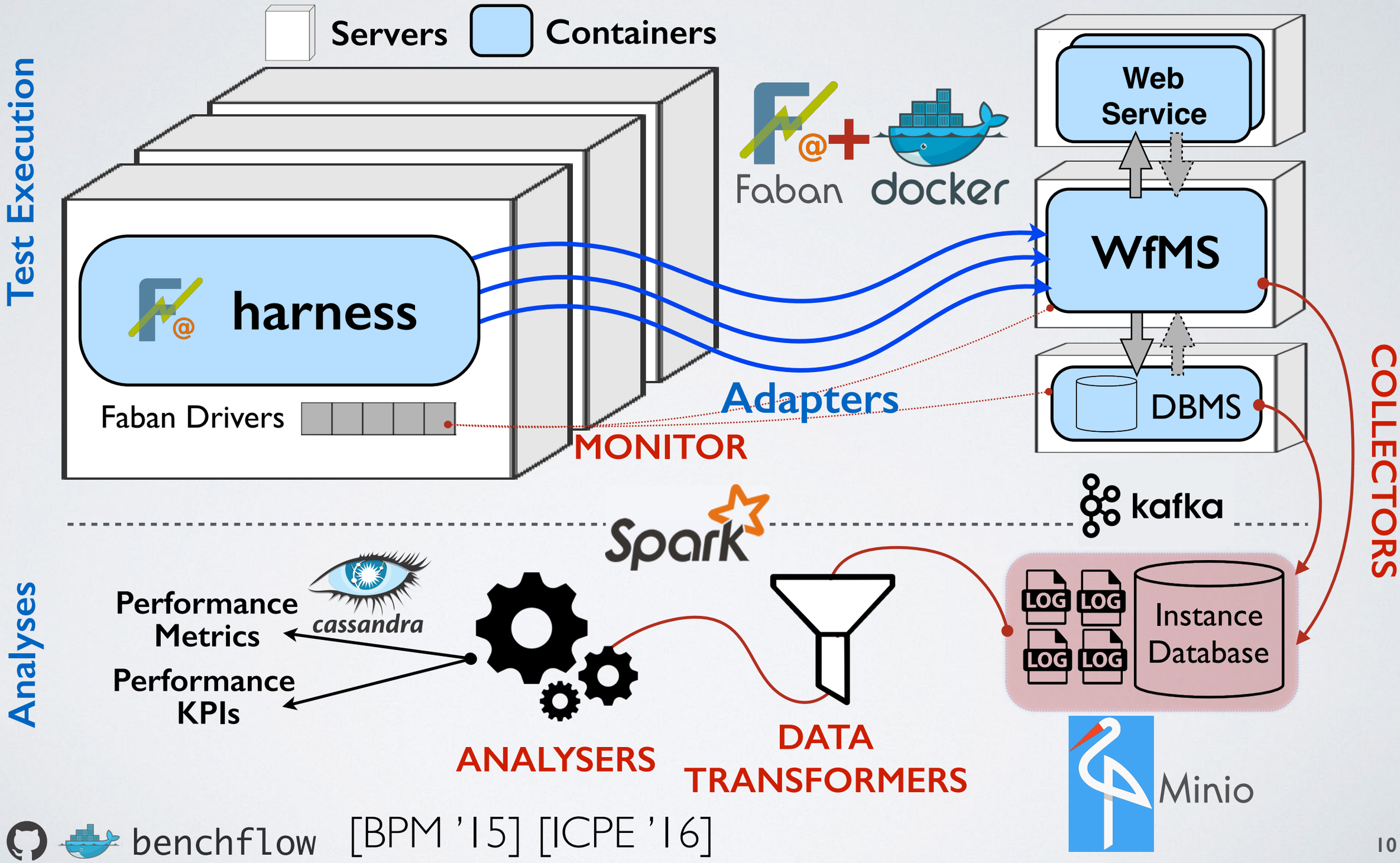
- Workflow & Construct:
  - Start Time
  - End Time
  - [Duration]

Availability of Execution State

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

BenchFlow Framework

architecture



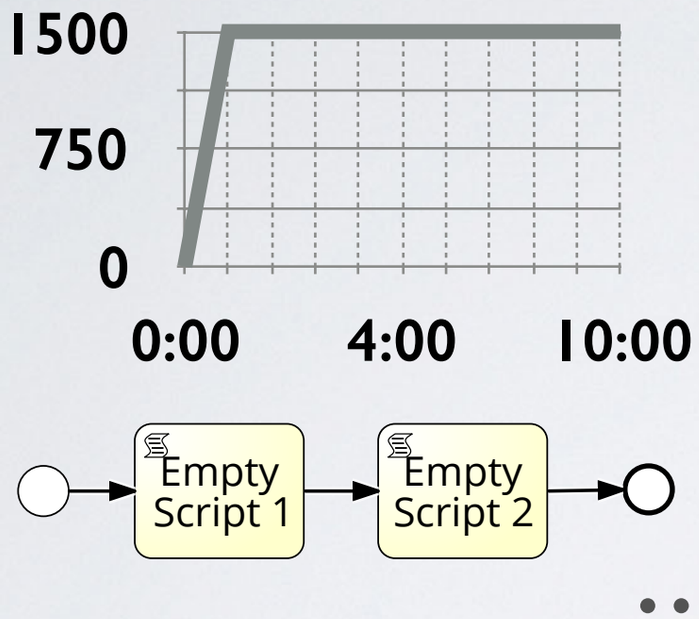
benchflow [BPM '15] [ICPE '16]

BenchFlow Framework

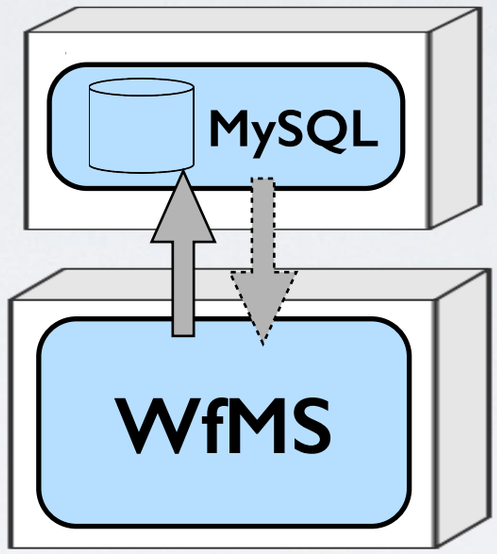
WfMSs' specific characteristics

- Manages the Automatic WfMS deployment;
- Provides a "plugin" mechanism to add new WfMSs;
- Performs automatic Process Models deployment;
- Collects Client and Server-side data and metrics;
- Automatically computes performance metrics and statistics.

Applications of the Methodology and Framework



Workload



3 WfMSs

- **Engine Level**
- **Process Level**
- **Environment**

Metrics

[CAiSE '16]

Micro-Benchmarking BPMN 2.0 Workflow Management Systems with Workflow Patterns.

Micro-Benchmarking with Workflow Patterns

research questions

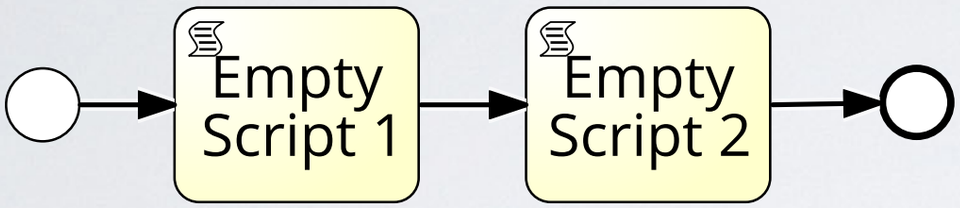
1. What is the impact of **individual** or a **mix** of workflow patterns on the performance of each one of the benchmarked BPMN 2.0 WfMSs?
2. Are there performance bottlenecks for the selected WfMSs?

***Selected three WfMSs:* popular open-source WfMSs actually used in industry**

Workload Mix

Micro-Benchmarking with Workflow Patterns

workload mix



Sequence Pattern [SEQ]

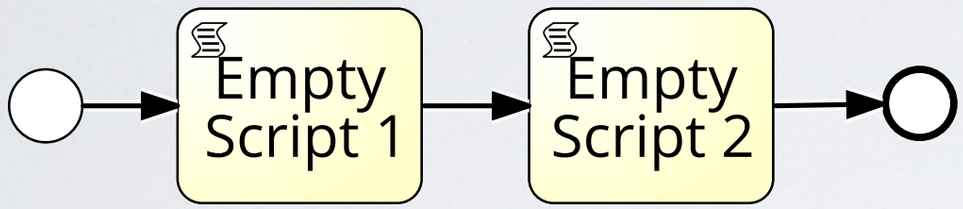
Pattern Reference:

- N. Russell et al, **Workflow Control-Flow Patterns: A Revised View**, 2006

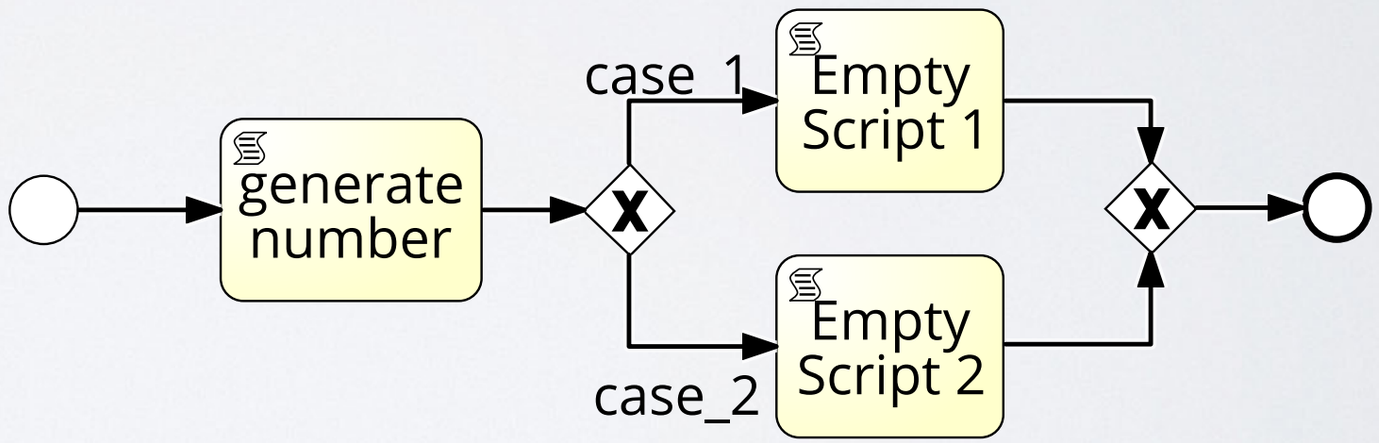
Workload Mix

Micro-Benchmarking with Workflow Patterns

workload mix



Sequence Pattern [SEQ]



Exclusive Choice and Simple Merge [EXC]

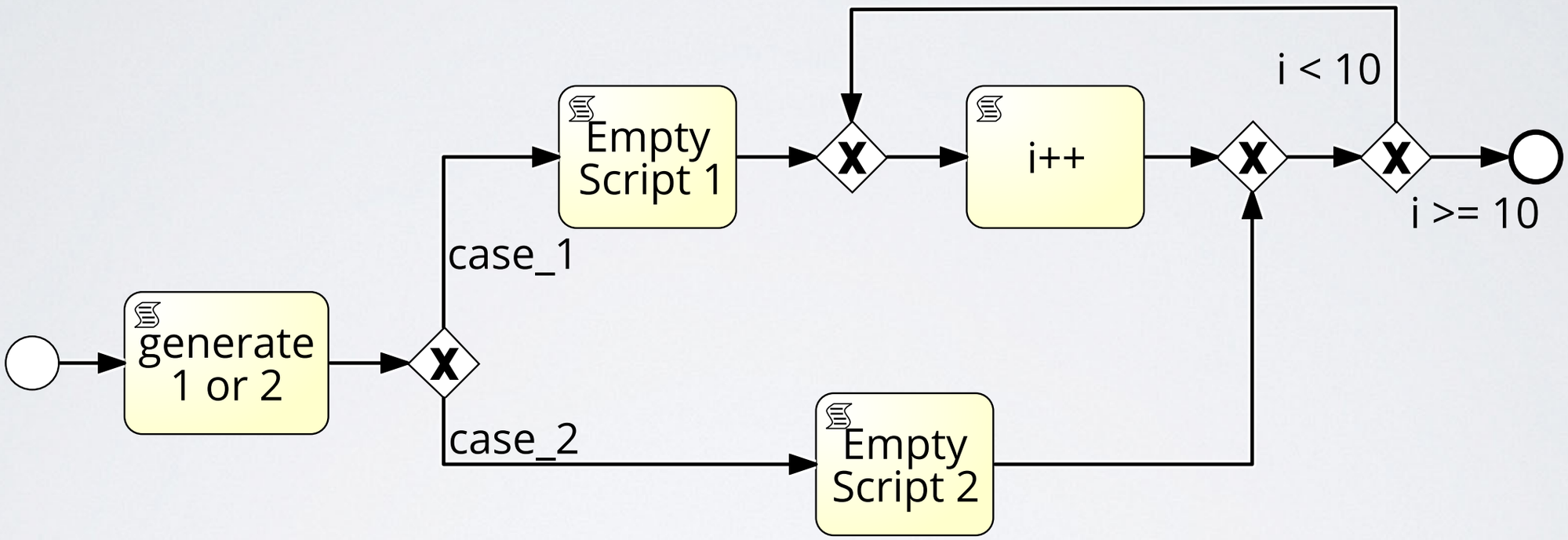
Pattern Reference:

- N. Russell et al, **Workflow Control-Flow Patterns: A Revised View**, 2006

Workload Mix

Micro-Benchmarking with Workflow Patterns

workload mix



Arbitrary Cycle [CYC]

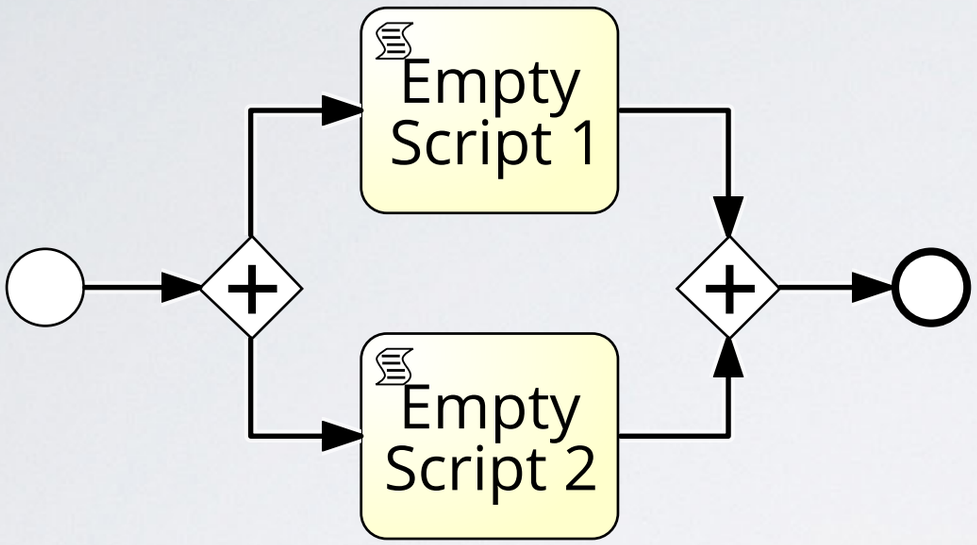
Pattern Reference:

- N. Russell et al, **Workflow Control-Flow Patterns: A Revised View**, 2006

Micro-Benchmarking with Workflow Patterns

workload mix

Workload Mix



Parallel Split and Synchronisation [PAR]

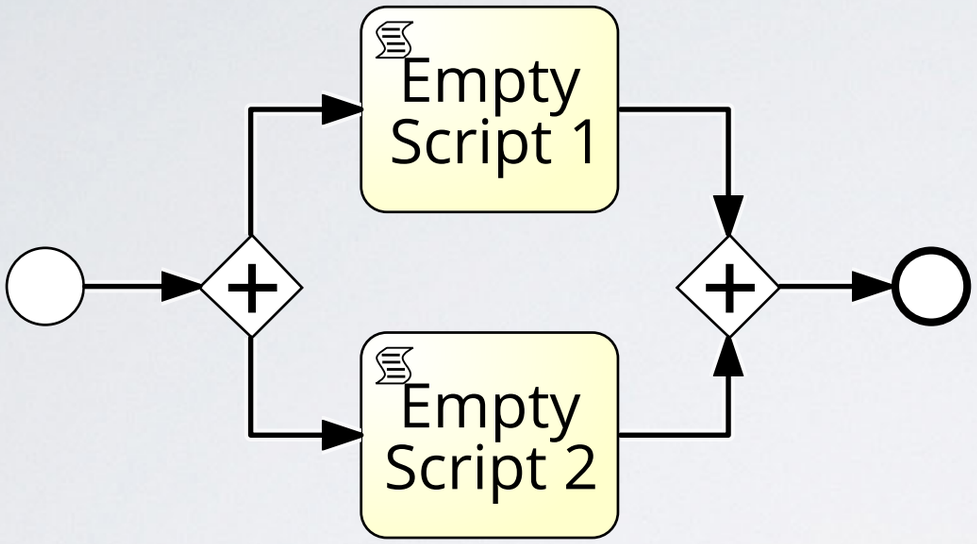
Pattern Reference:

- N. Russell et al, **Workflow Control-Flow Patterns: A Revised View**, 2006

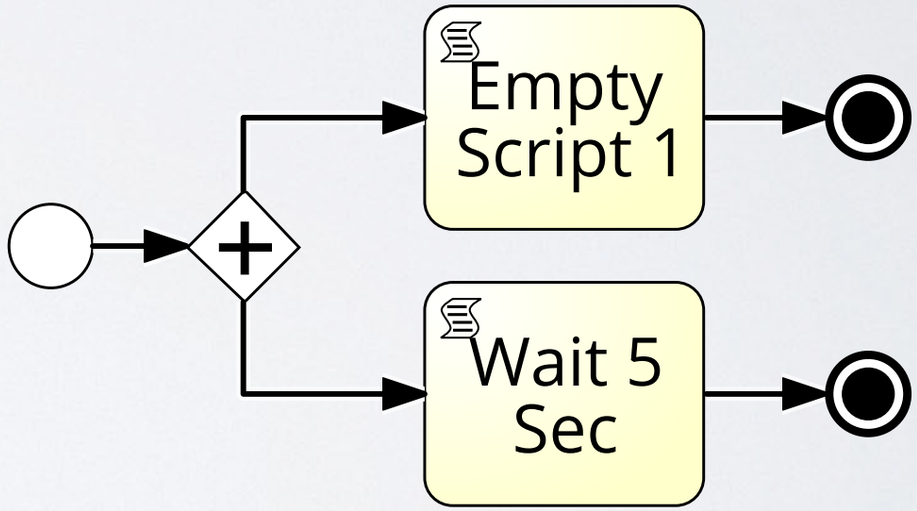
Workload Mix

Micro-Benchmarking with Workflow Patterns

workload mix



Parallel Split and Synchronisation [PAR]



Explicit Termination Pattern [EXT]

Pattern Reference:

- N. Russell et al, **Workflow Control-Flow Patterns: A Revised View**, 2006

Micro-Benchmarking with Workflow Patterns

workload mix design decisions

1. Maximise the simplicity of the model expressing the workflow pattern

2. Omit the interactions with external systems by implementing all tasks as script tasks

Micro-Benchmarking with Workflow Patterns

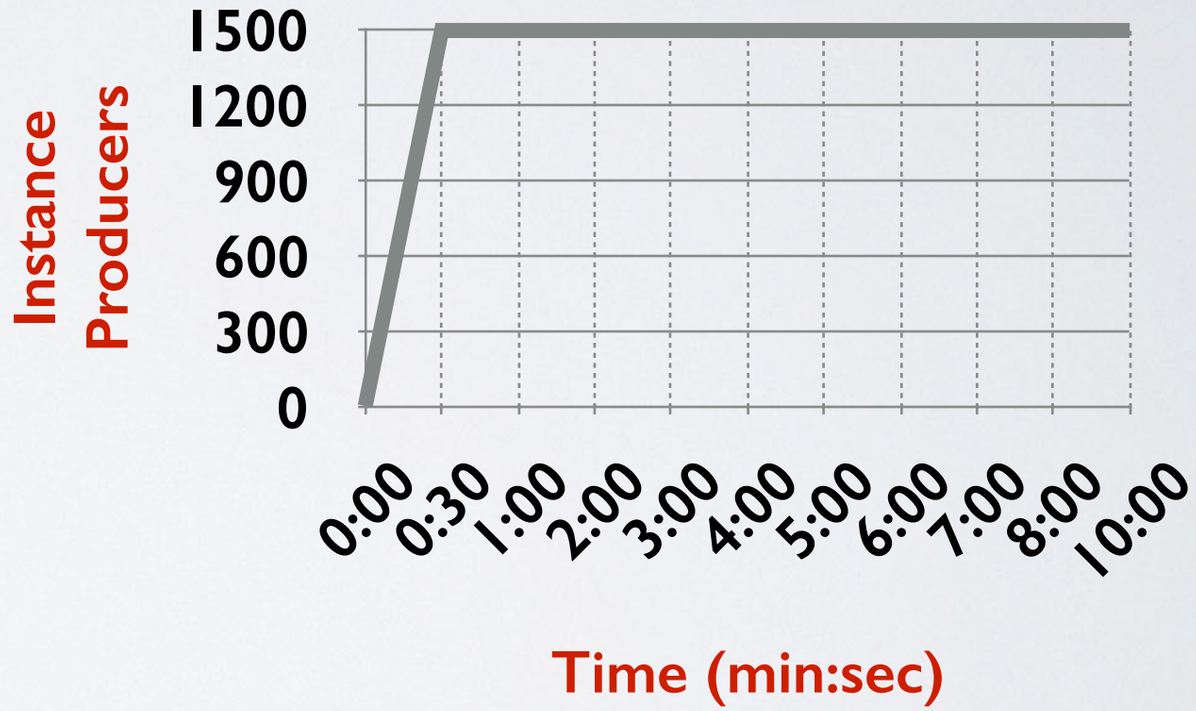
load function

Load Functions

Test Type → Load test

# of Instance Producers	u
Think Time	1 sec
Max User req/sec (r)	1
Load Time (T_l)	10 min
Ramp-Up Period (T_r)	30 sec
Connection Time-out (T_o)	20 sec

Example for *u=1500*

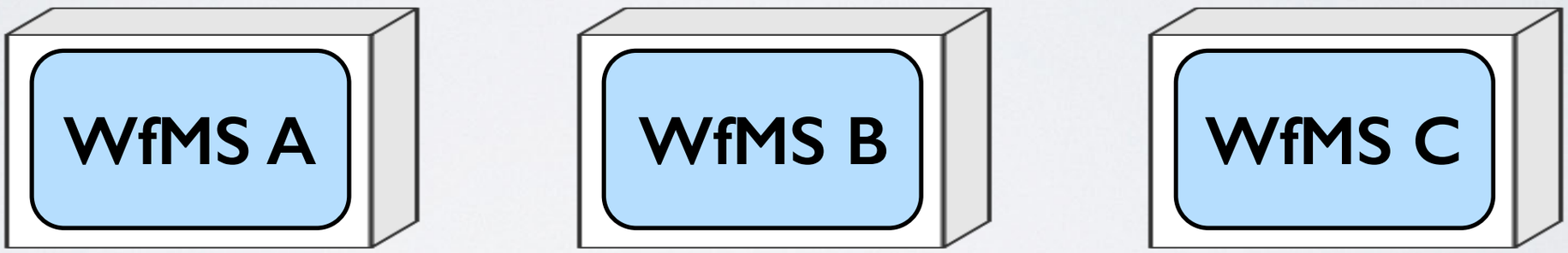


Micro-Benchmarking with Workflow Patterns

Configurations

WfMS

configurations: WfMS environments



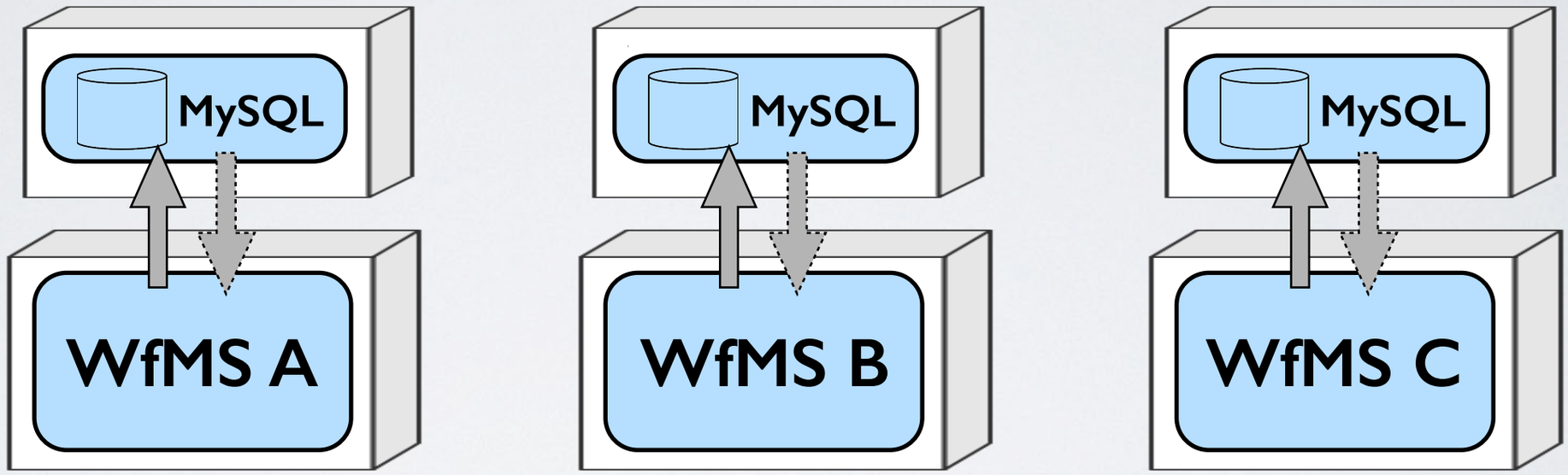
Configurations

WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS environments

MySQL: Community Server 5.6.26



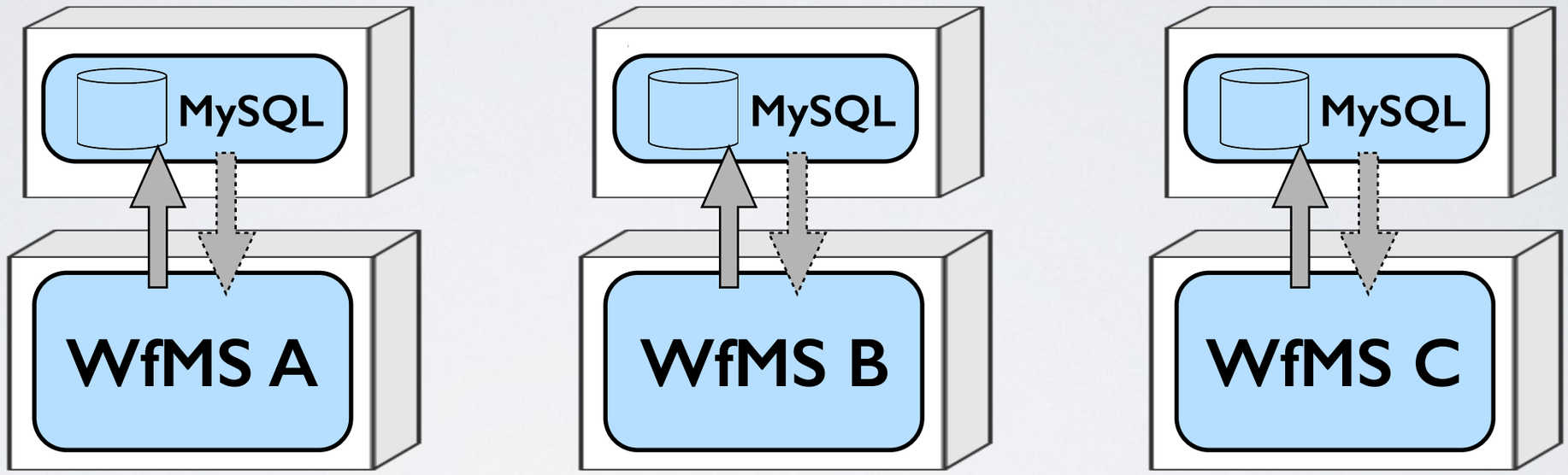
Configurations

WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS environments

MySQL: Community Server 5.6.26



O.S.: Ubuntu 14.04.01
J.V.M.: Oracle Serv. 7u79

Ubuntu 14.04.01
Oracle Serv. 7u79

Ubuntu 14.04.01
Oracle Serv. 7u79

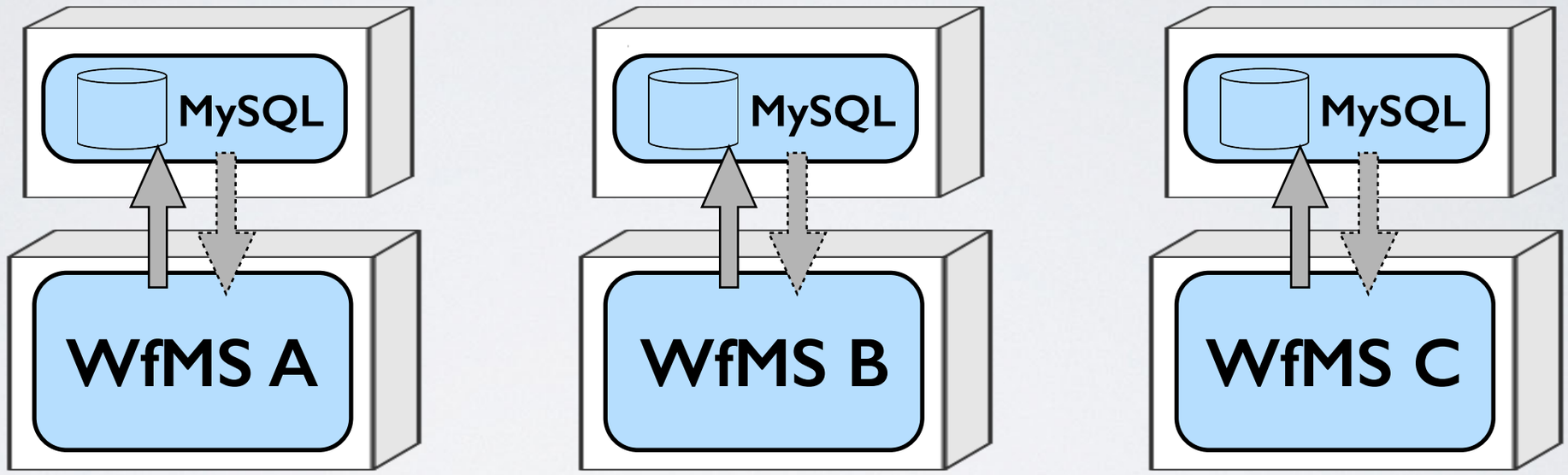
Configurations

WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS environments

MySQL: Community Server 5.6.26



	WfMS A	WfMS B	WfMS C
O.S.:	Ubuntu 14.04.01	Ubuntu 14.04.01	Ubuntu 14.04.01
J.V.M.:	Oracle Serv. 7u79	Oracle Serv. 7u79	Oracle Serv. 7u79
App. Server:	Ap. Tomcat 7.0.62	Ap. Tomcat 7.0.62	Wildfly 8.1.0. Final

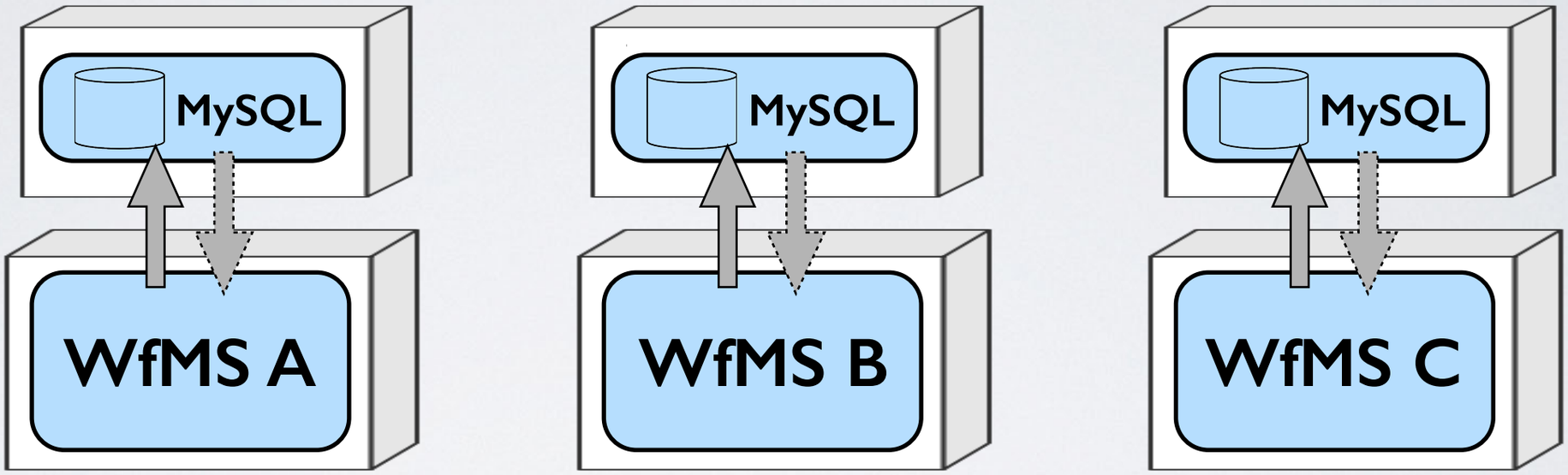
Configurations

WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS environments

MySQL: Community Server 5.6.26



	WfMS A	WfMS B	WfMS C
O.S.:	Ubuntu 14.04.01	Ubuntu 14.04.01	Ubuntu 14.04.01
J.V.M.:	Oracle Serv. 7u79	Oracle Serv. 7u79	Oracle Serv. 7u79
App. Server:	Ap. Tomcat 7.0.62	Ap. Tomcat 7.0.62	Wildfly 8.1.0. Final

Max Java Heap: 32GB

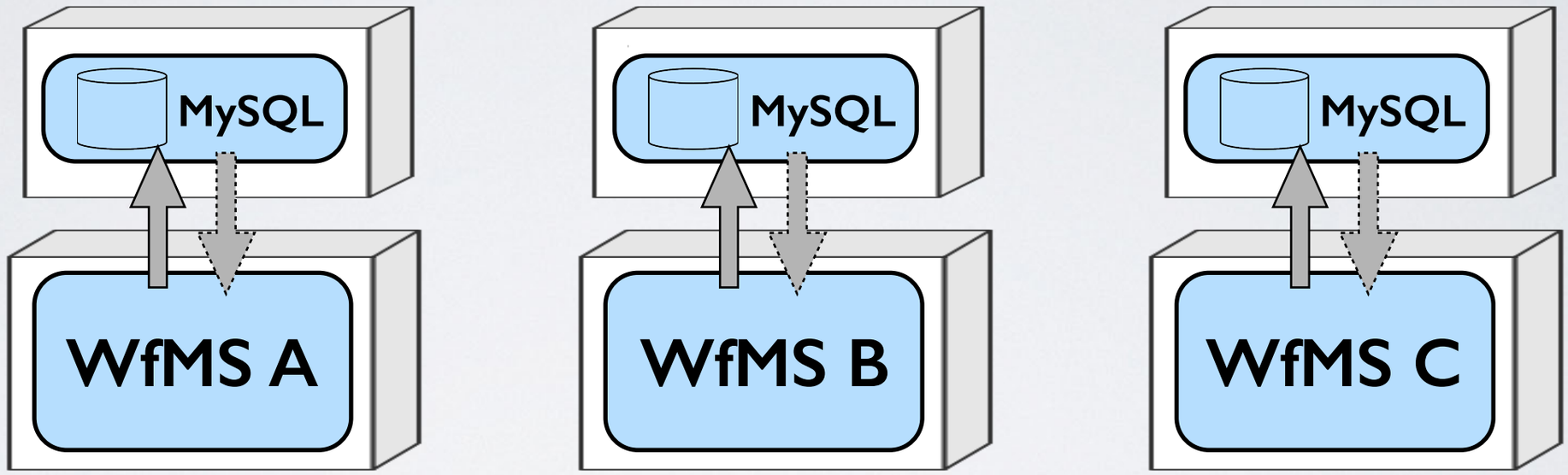
Configurations

WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS environments

MySQL: Community Server 5.6.26



	WfMS A	WfMS B	WfMS C
O.S.:	Ubuntu 14.04.01	Ubuntu 14.04.01	Ubuntu 14.04.01
J.V.M.:	Oracle Serv. 7u79	Oracle Serv. 7u79	Oracle Serv. 7u79
App. Server:	Ap. Tomcat 7.0.62	Ap. Tomcat 7.0.62	Wildfly 8.1.0. Final

Max Java Heap: 32GB

Max DB Connections Number: 100

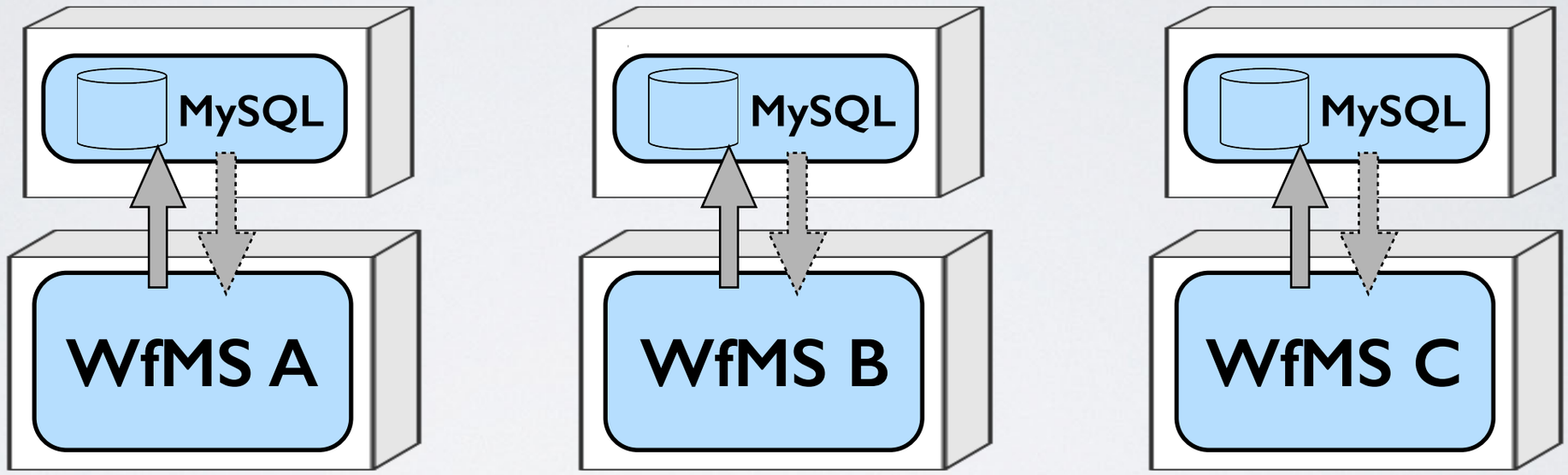
Configurations

WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS environments

MySQL: Community Server 5.6.26



	WfMS A	WfMS B	WfMS C
O.S.:	Ubuntu 14.04.01	Ubuntu 14.04.01	Ubuntu 14.04.01
J.V.M.:	Oracle Serv. 7u79	Oracle Serv. 7u79	Oracle Serv. 7u79
App. Server:	Ap. Tomcat 7.0.62	Ap. Tomcat 7.0.62	Wildfly 8.1.0. Final

Max Java Heap: 32GB

Max DB Connections Number: 100

WfMS's Containers: official ones, if available on DockerHub

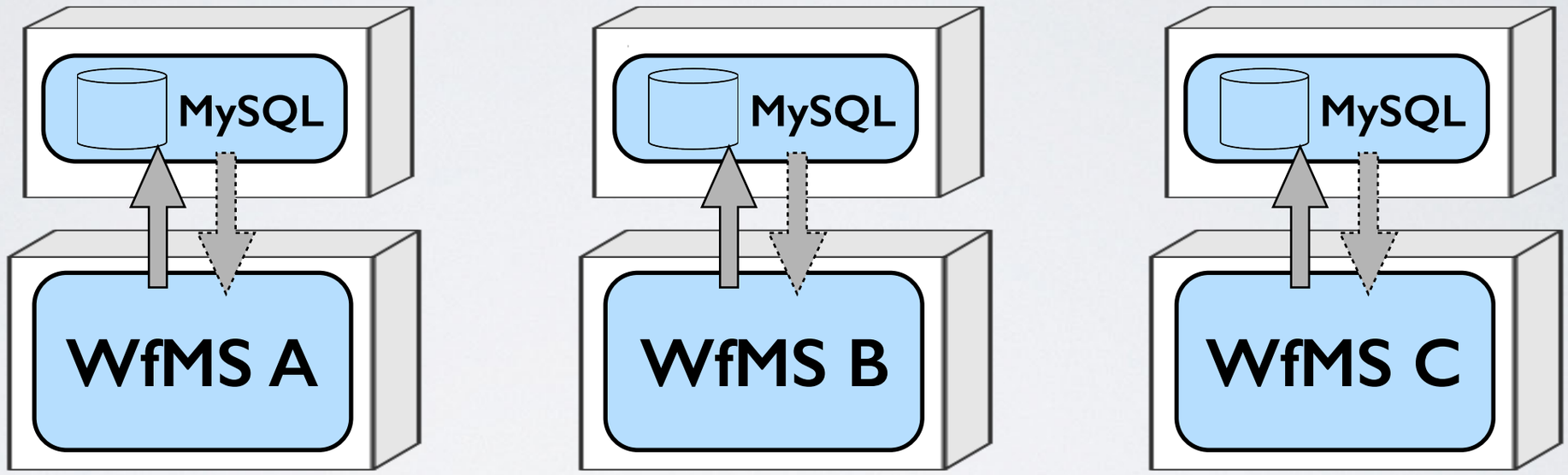
Configurations

WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS environments

MySQL: Community Server 5.6.26



	WfMS A	WfMS B	WfMS C
O.S.:	Ubuntu 14.04.01	Ubuntu 14.04.01	Ubuntu 14.04.01
J.V.M.:	Oracle Serv. 7u79	Oracle Serv. 7u79	Oracle Serv. 7u79
App. Server:	Ap. Tomcat 7.0.62	Ap. Tomcat 7.0.62	Wildfly 8.1.0. Final

Max Java Heap: 32GB

Max DB Connections Number: 100

WfMS's Containers: official ones, if available on DockerHub

WfMS's Configuration: as suggested by vendor's documentation

Configurations
WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS deployment

Load Drivers

CPU	64 Cores @ 1400 MHz
RAM	128 GB

WfMS

CPU	12 Cores @ 800 MHz
RAM	64 GB

DBMS

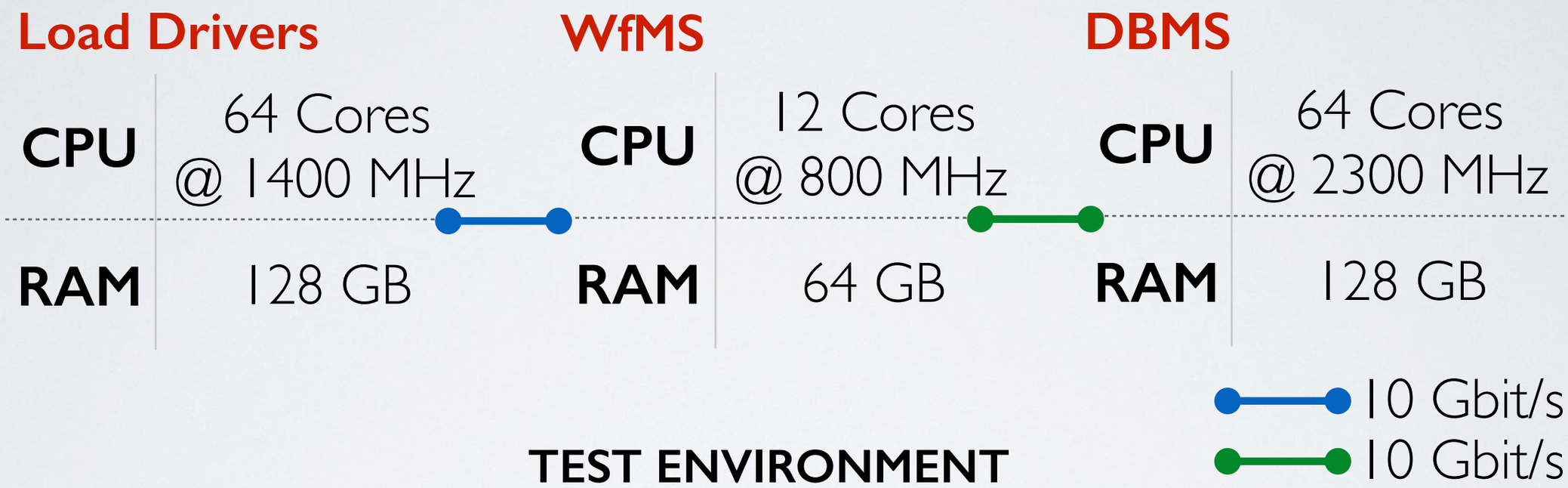
CPU	64 Cores @ 2300 MHz
RAM	128 GB

TEST ENVIRONMENT

Configurations
WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS deployment

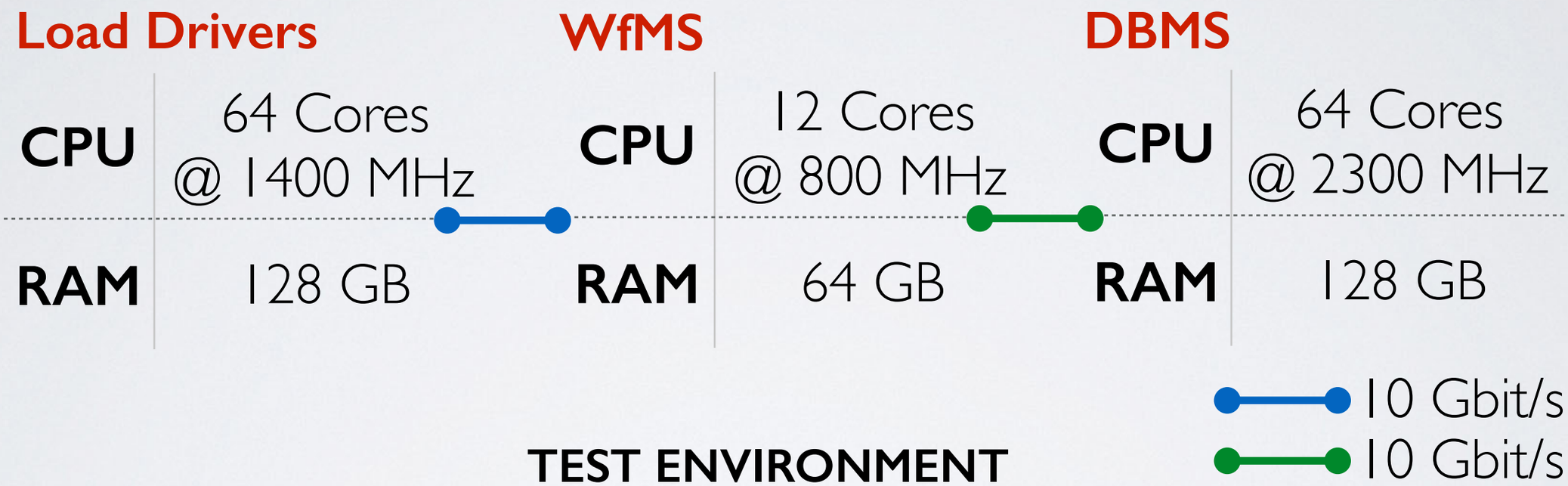


TEST ENVIRONMENT

Configurations
WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS deployment



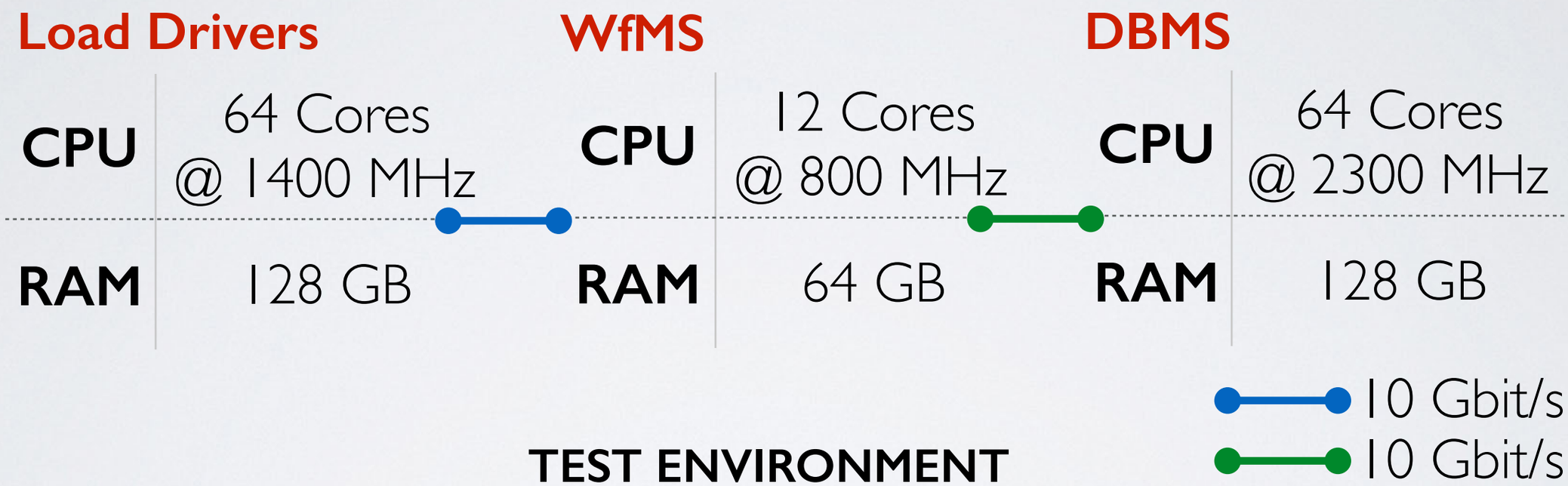
TEST ENVIRONMENT

O.S.: Ubuntu 14.04.3 LTS | Ubuntu 14.04.3 LTS | Ubuntu 14.04.3 LTS

Configurations
WfMS

Micro-Benchmarking with Workflow Patterns

configurations: WfMS deployment



Micro-Benchmarking with Workflow Patterns

performed experiments

Individual patterns

1. 1500 (max) concurrent Instance Producers starting each pattern [SEQ, EXC, CYC, EXT, PAR]

Mix of patterns

2. 1500 concurrent Instance Producers starting an equal number of pattern instances (20% mix of [SEQ, EXC, CYC, EXT, PAR])

Micro-Benchmarking with Workflow Patterns

performed experiments

Individual patterns

1. 1500 (max) concurrent Instance Producers starting each pattern [SEQ, EXC, CYC, EXT, PAR]

Mix of patterns

2. 1500 concurrent Instance Producers starting an equal number of pattern instances (20% mix of [SEQ, EXC, CYC, EXT, PAR])

Three runs for each execution of the experiments

Micro-Benchmarking with Workflow Patterns

performed experiments

Individual patterns

1. 1500 (max) concurrent Instance Producers starting each pattern [SEQ, EXC, CYC, EXT, PAR]

Mix of patterns

2. 1500 concurrent Instance Producers starting an equal number of pattern instances (20% mix of [SEQ, EXC, CYC, EXT, PAR])

Three runs for each execution of the experiments

The data collected in the first minute of the execution has not been included in the analysis

Micro-Benchmarking with Workflow Patterns

performed experiments

Individual patterns

1. 1500 (max) concurrent Instance Producers starting each pattern [SEQ, EXC, CYC, EXT, PAR]

Mix of patterns

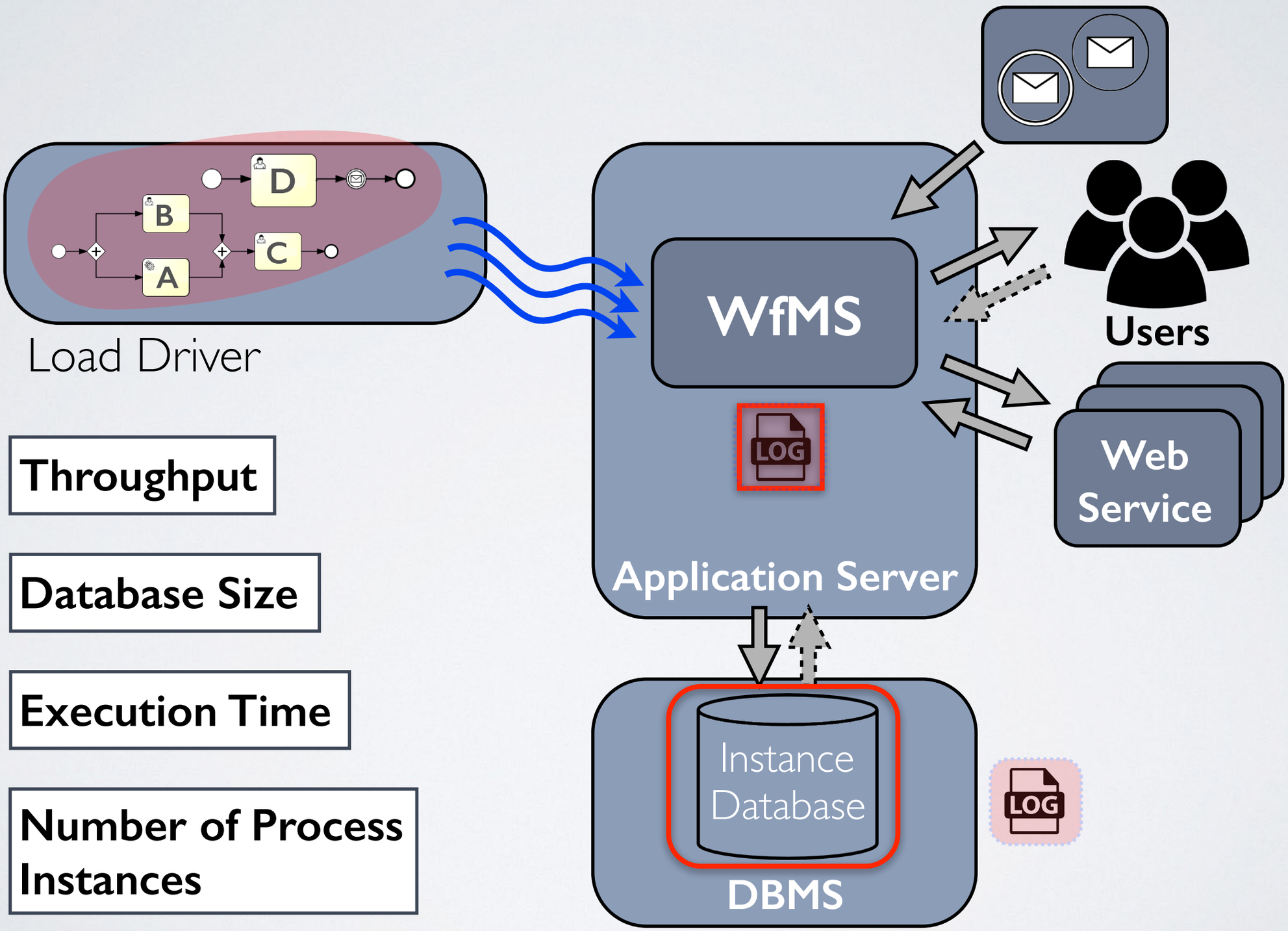
2. 1500 concurrent Instance Producers starting an equal number of pattern instances (20% mix of [SEQ, EXC, CYC, EXT, PAR])

Three runs for each execution of the experiments

The data collected in the first minute of the execution has not been included in the analysis

Computed Metrics and Statistics

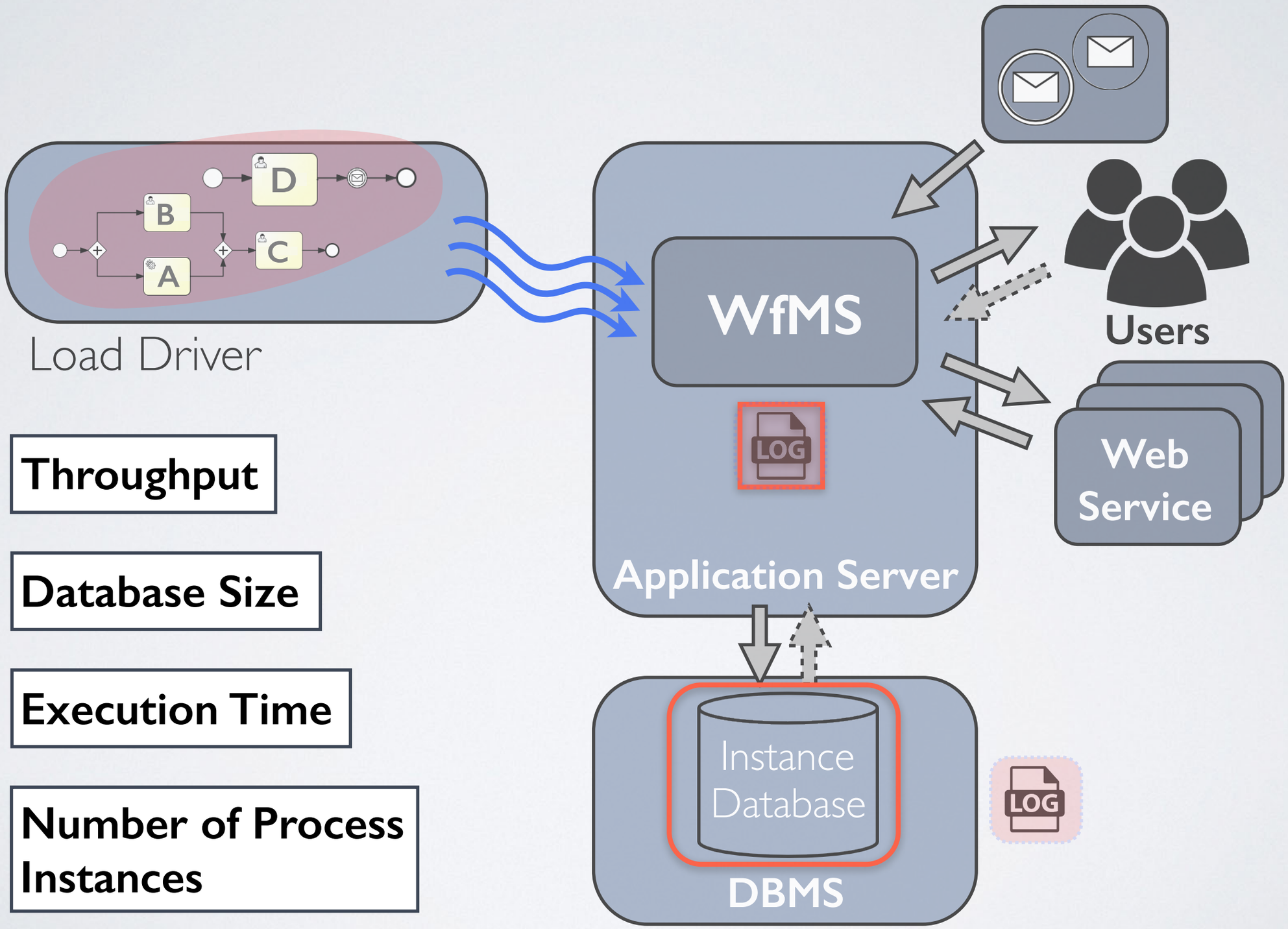
engine level metrics



- **Throughput**
- **Database Size**
- **Execution Time**
- **Number of Process Instances**
- ...

Computed Metrics and Statistics

engine level metrics



- Throughput
- Database Size
- Execution Time
- Number of Process Instances
- ...

Micro-Benchmarking with Workflow Patterns

individual patterns: throughput (bp/sec)

	SEQ	EXC	EXT	PAR	CYC
WfMS A	1456.79	1417.17	1455.68	1433.12	327.99 *u=600*
WfMS B	1447.66	1436.03	1426.35	1429.65	644.02 *u=800*
WfMS C	63.31	49.04	0.38	48.89	13.46 *u=1500*

Micro-Benchmarking with Workflow Patterns

individual patterns: WfMS C Scalability for SEQ pattern

WfMS C uses a synchronous instantiation API, load drivers must wait for workflows to end before starting new ones

Micro-Benchmarking with Workflow Patterns

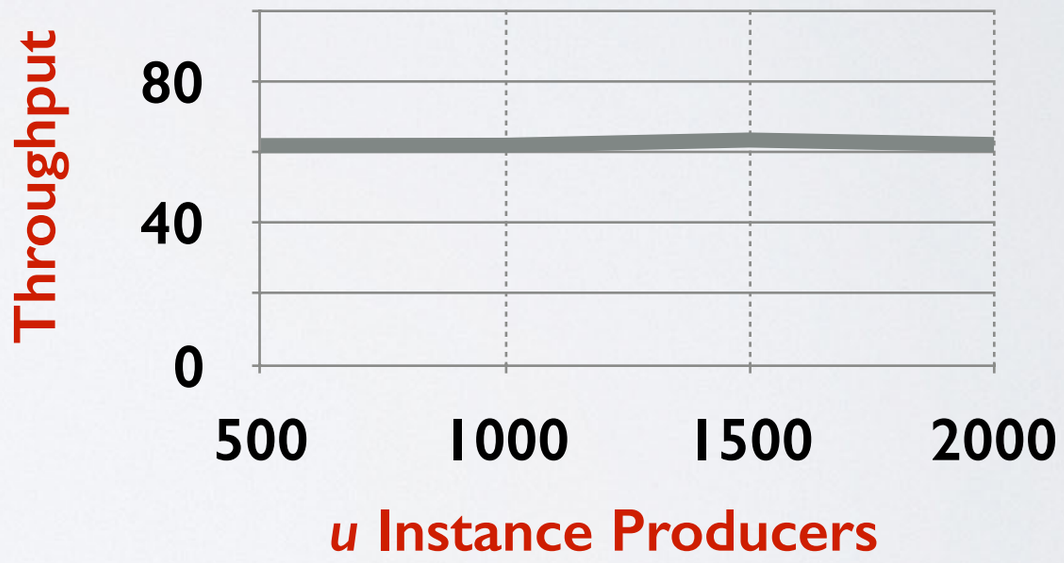
individual patterns: WfMS C Scalability for SEQ pattern

WfMS C uses a synchronous instantiation API, load drivers must wait for workflows to end before starting new ones

Micro-Benchmarking with Workflow Patterns

individual patterns: WfMS C Scalability for SEQ pattern

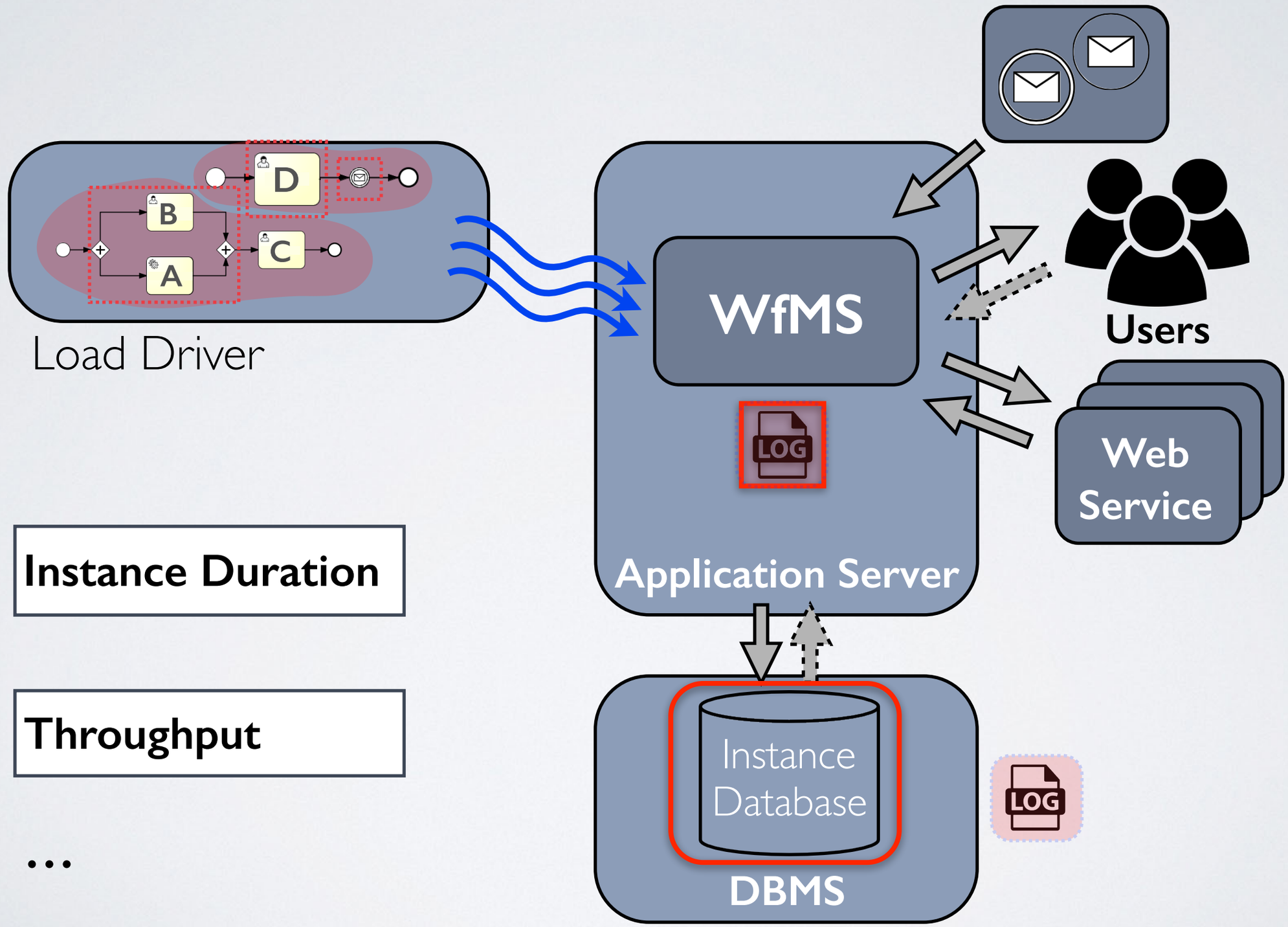
WfMS C uses a synchronous instantiation API, load drivers must wait for workflows to end before starting new ones



We have found a bottleneck! (~60 bp/sec)

Computed Metrics and Statistics

process level metrics



Per Process Definition

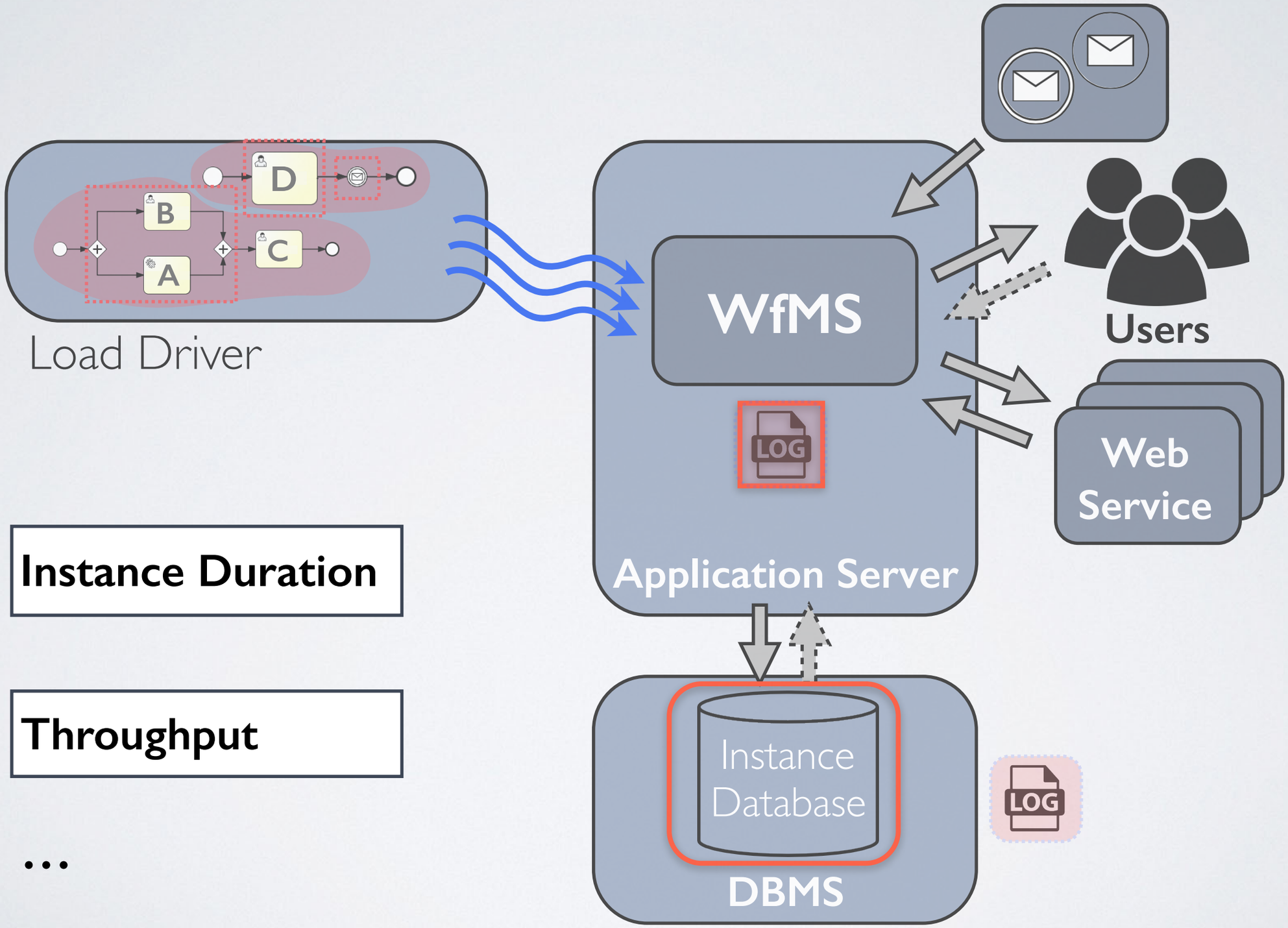
- **Instance Duration**
- **Throughput**
- …

Computed Metrics and Statistics

process level metrics

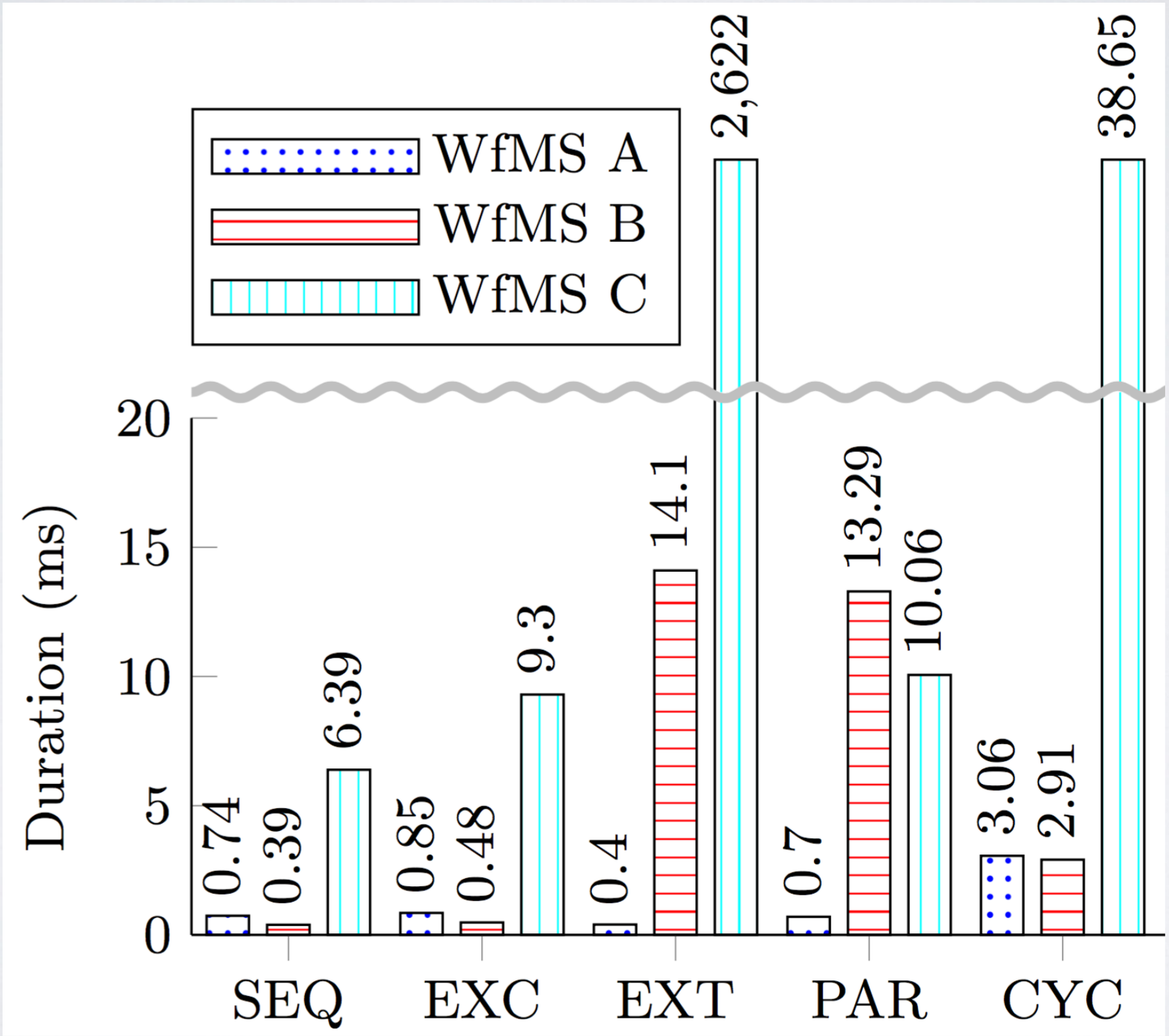
Per Process Definition

- Instance Duration
- Throughput
- …



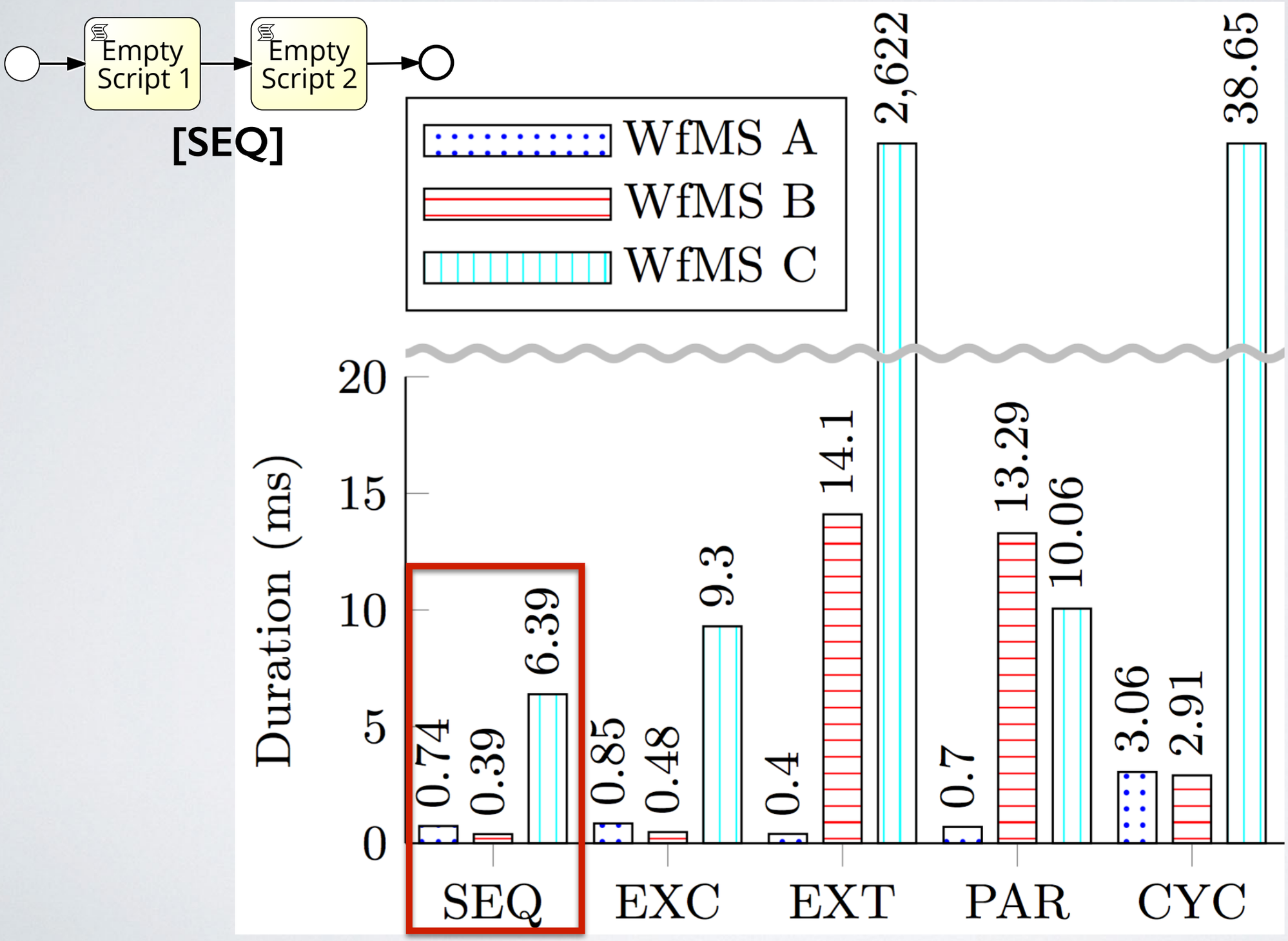
Micro-Benchmarking with Workflow Patterns

individual patterns: mean (instance duration time)



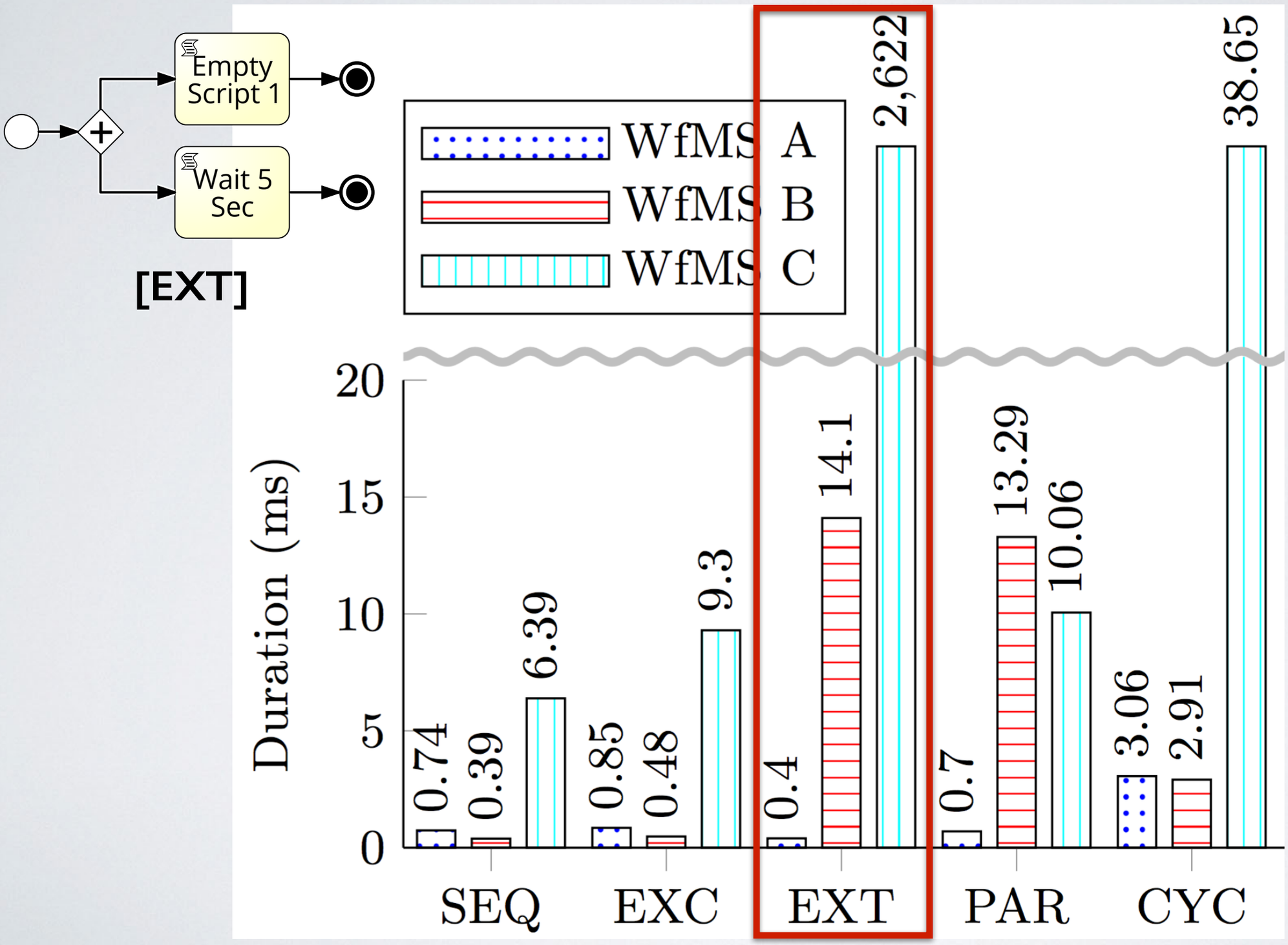
Micro-Benchmarking with Workflow Patterns

individual patterns: mean (instance duration time)



Micro-Benchmarking with Workflow Patterns

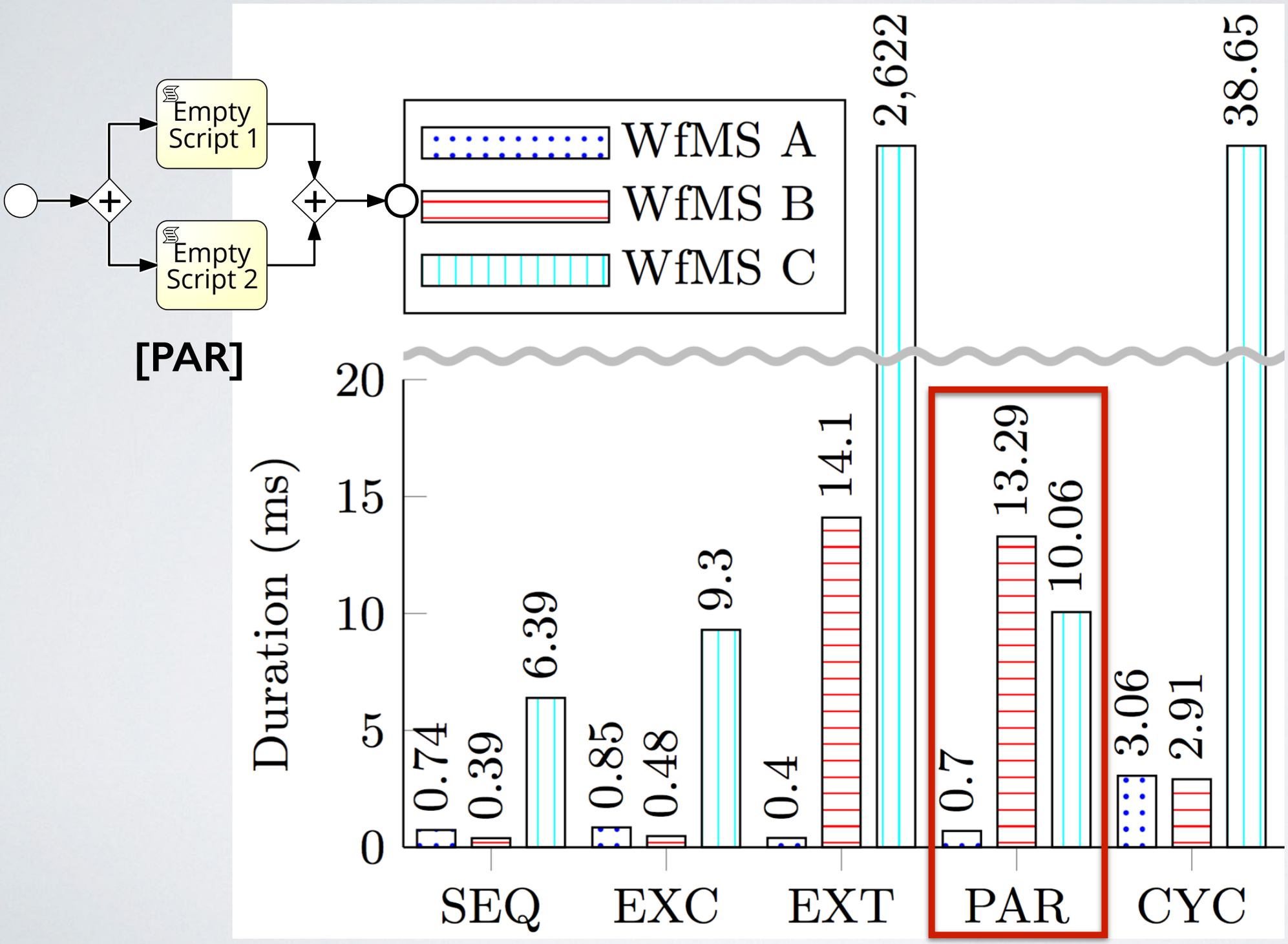
individual patterns: mean (instance duration time)



Micro-Benchmarking with Workflow Patterns

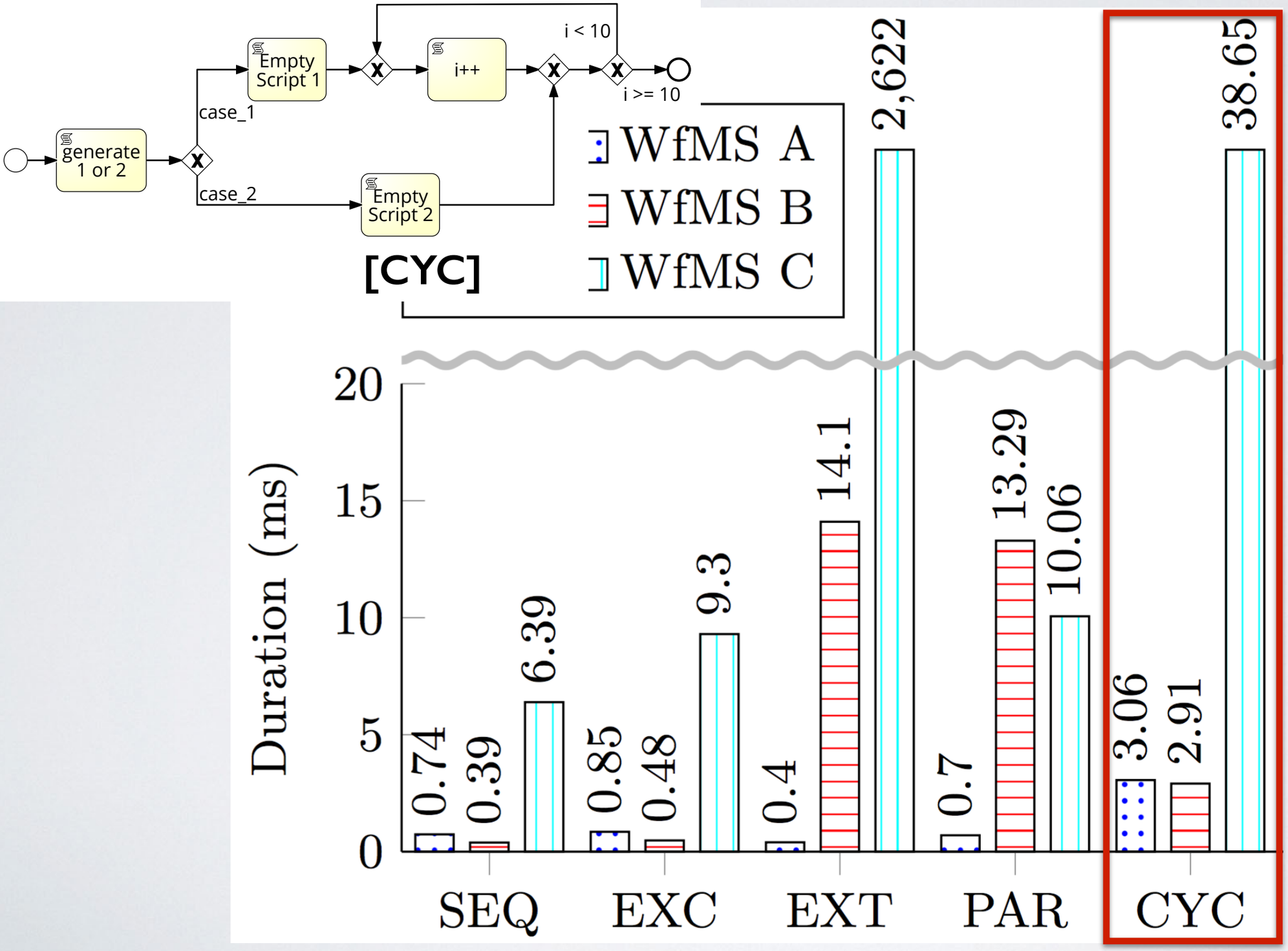
individual patterns: mean (instance duration time)

[PAR]



Micro-Benchmarking with Workflow Patterns

individual patterns: mean (instance duration time)

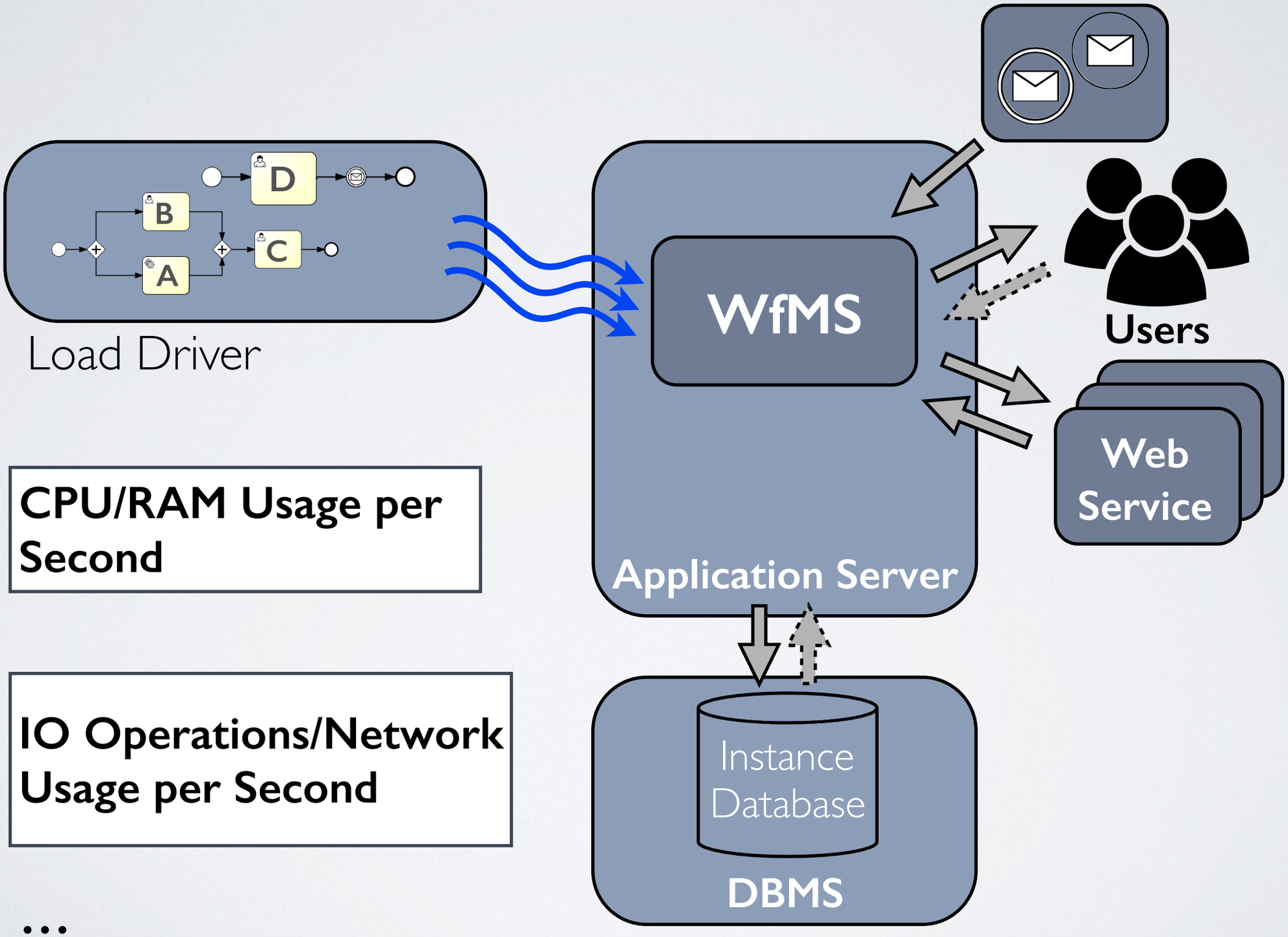


CYC Instance Producers:

u=600

Computed Metrics and Statistics

environment metrics



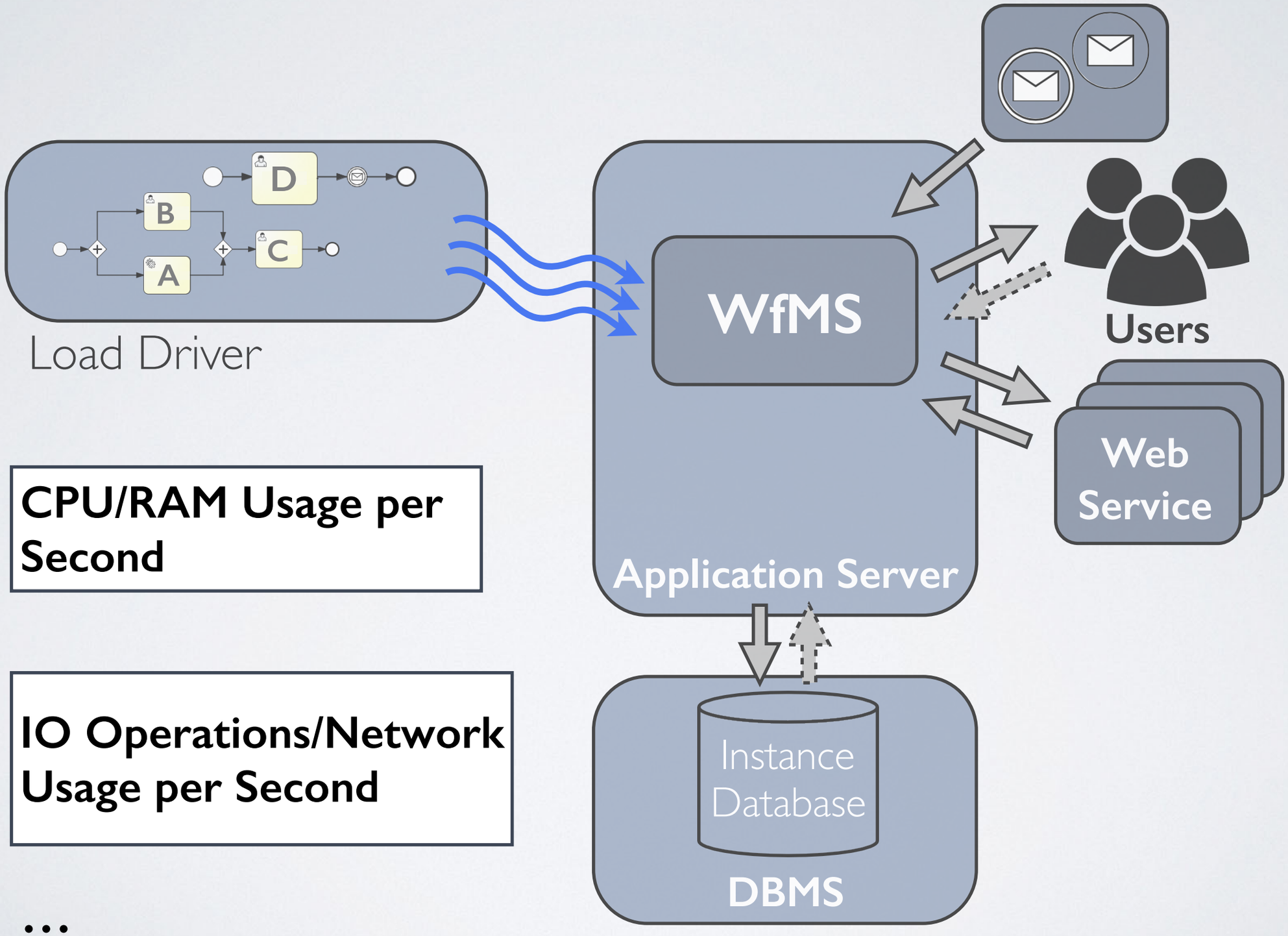
CPU/RAM Usage per Second

IO Operations/Network Usage per Second

...

Computed Metrics and Statistics

environment metrics



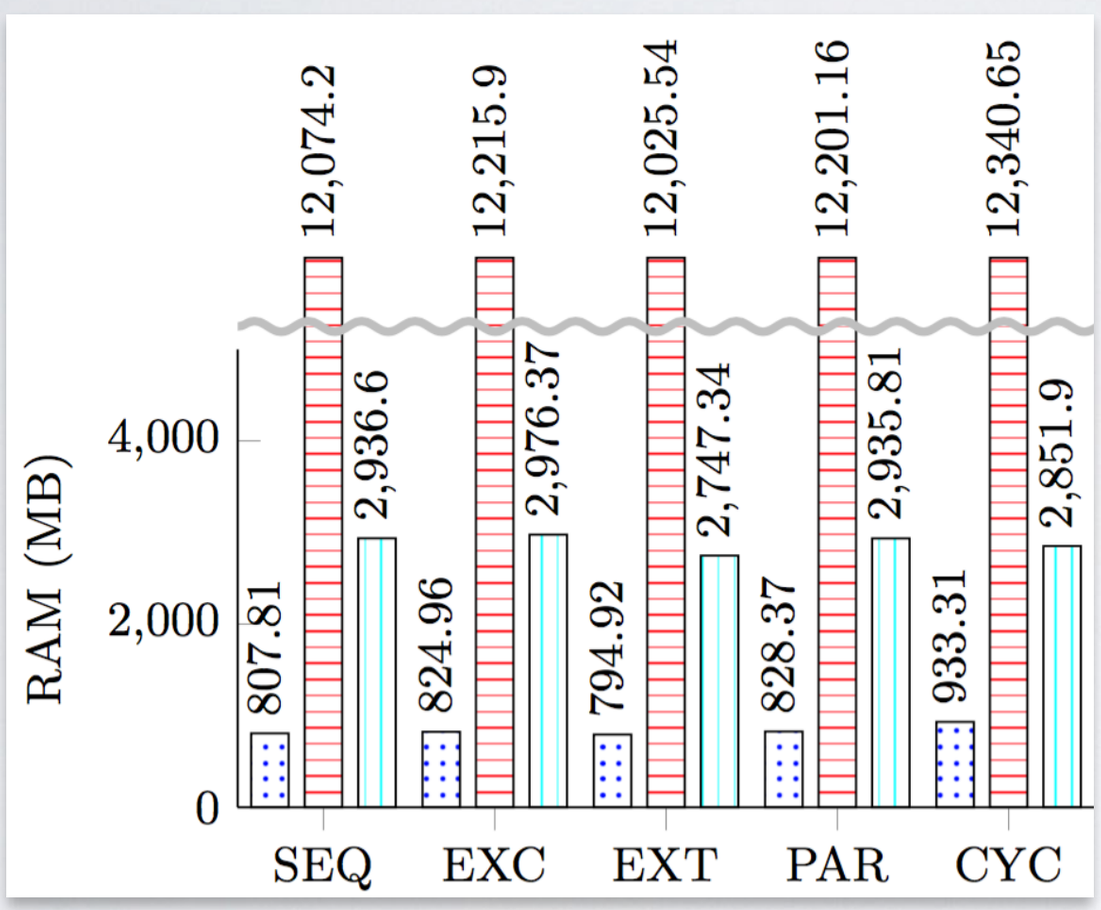
CPU/RAM Usage per Second

IO Operations/Network Usage per Second

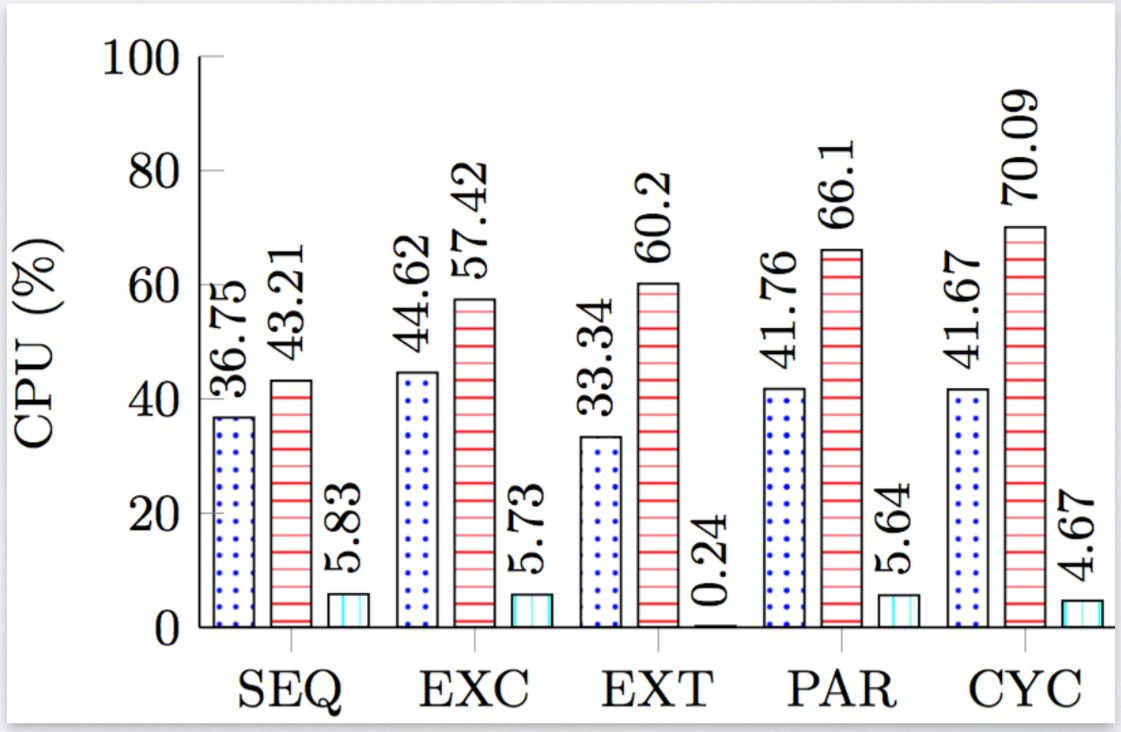
...

Micro-Benchmarking with Workflow Patterns

individual patterns: mean (RAM) and mean (CPU) usage



RAM



CPU

Micro-Benchmarking with Workflow Patterns

performed experiments

Individual patterns

1. 1500 (max) concurrent Instance Producers starting each pattern [SEQ, EXC, CYC, EXT, PAR]

Mix of patterns

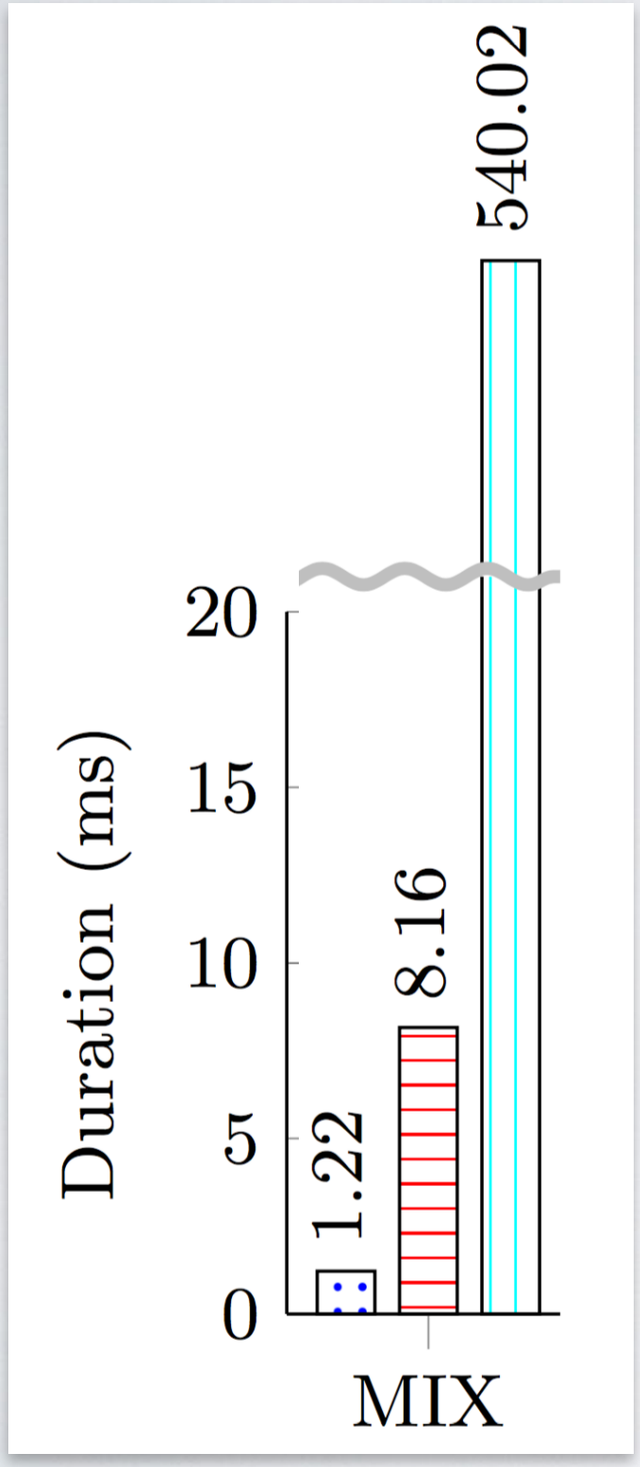
2. 1500 concurrent Instance Producers starting an equal number of pattern instances (20% mix of [SEQ, EXC, CYC, EXT, PAR])

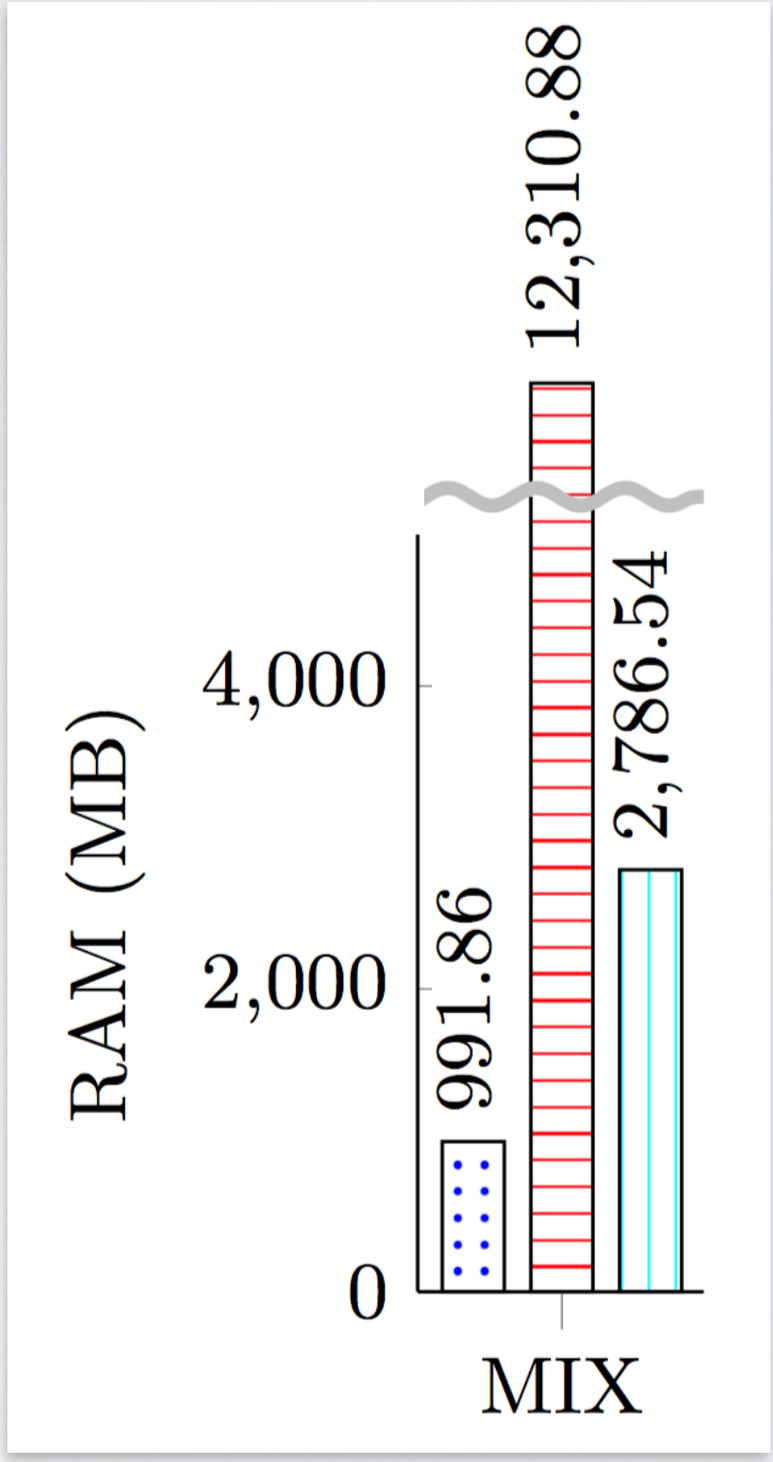
Three runs for each execution of the experiments

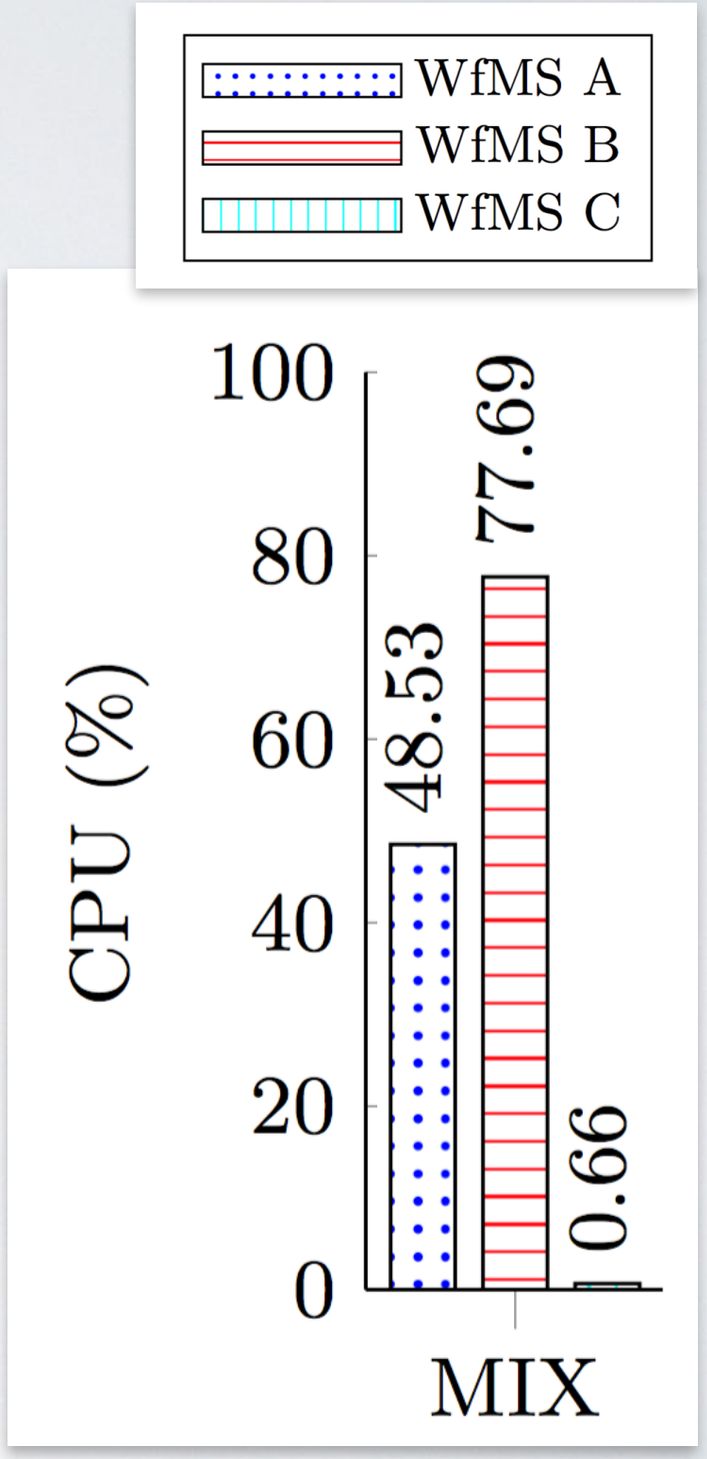
The data collected in the first minute of the execution has not been included in the analysis

Micro-Benchmarking with Workflow Patterns

mix of patterns: duration, RAM, CPU







Micro-Benchmarking with Workflow Patterns

summary

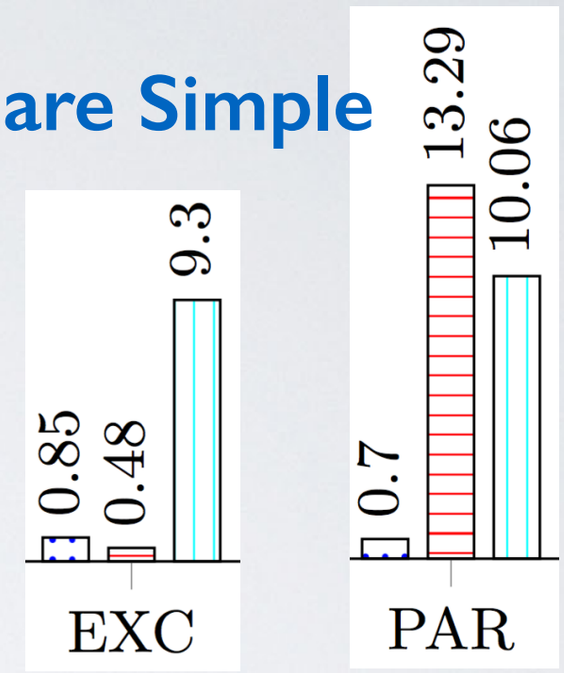
Even though the Executed Workflows are Simple

Micro-Benchmarking with Workflow Patterns

summary

Even though the Executed Workflows are Simple

There are relevant differences in workflows' duration and throughput among the WfMSs

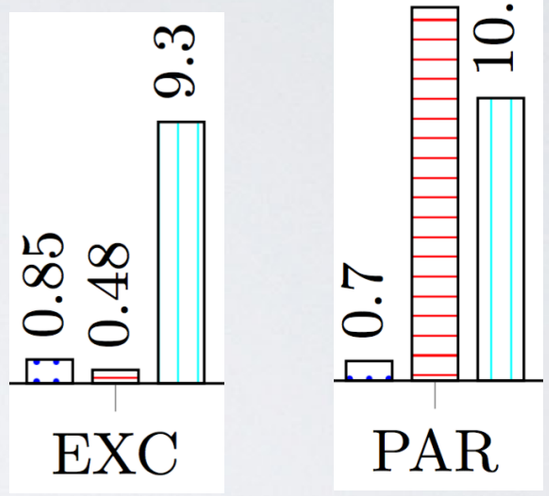


Micro-Benchmarking with Workflow Patterns

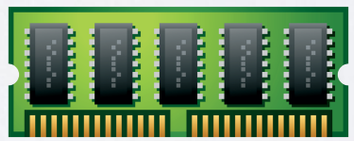
summary

Even though the Executed Workflows are Simple

There are relevant differences in workflows' duration and throughput among the WfMSs



There are relevant differences in resource usage among WfMSs



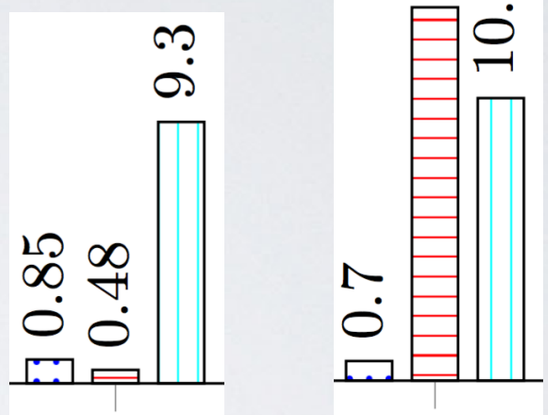
RAM

CPU

Micro-Benchmarking with Workflow Patterns

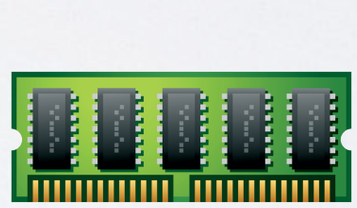
summary

Even though the Executed Workflows are Simple



There are relevant differences in workflows' duration and throughput among the WfMSs

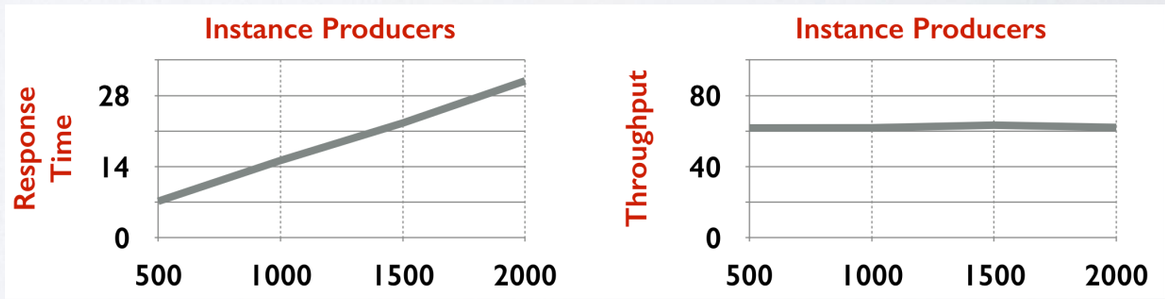
There are relevant differences in resource usage among WfMSs



RAM

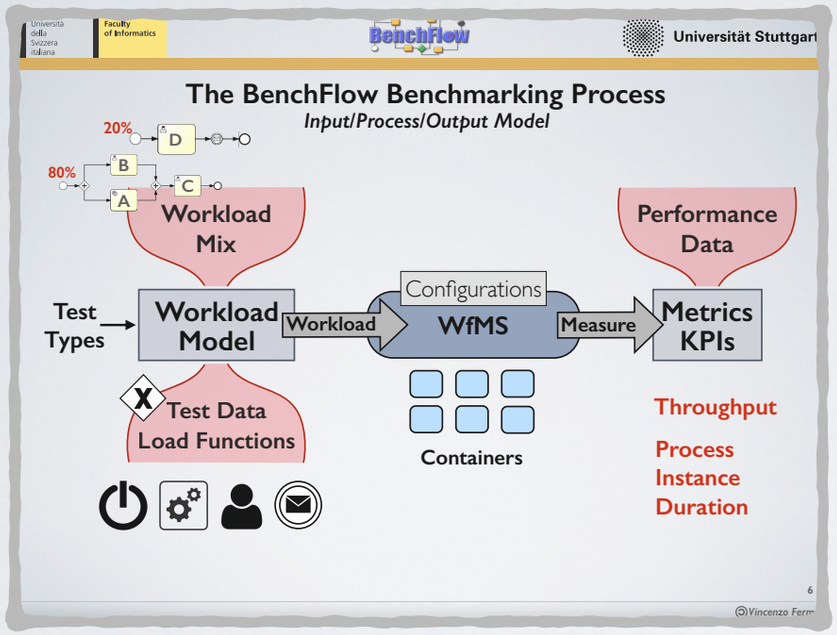
CPU

One WfMS has bottlenecks



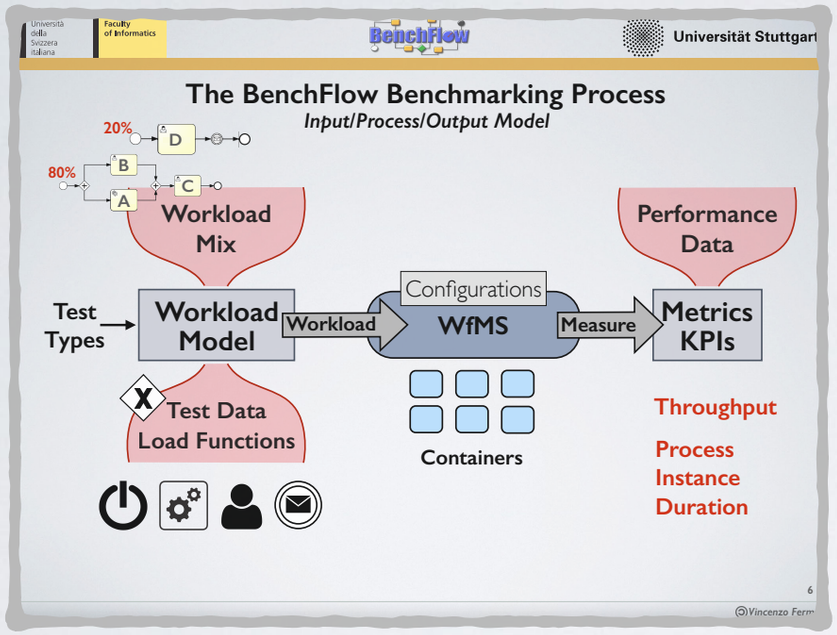
Highlights

Highlights

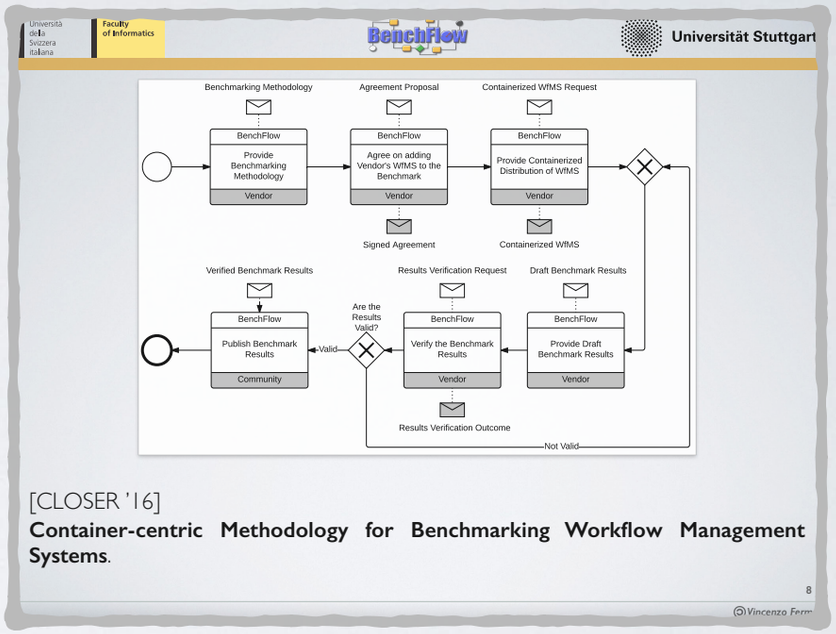


Benchmarking Process

Highlights



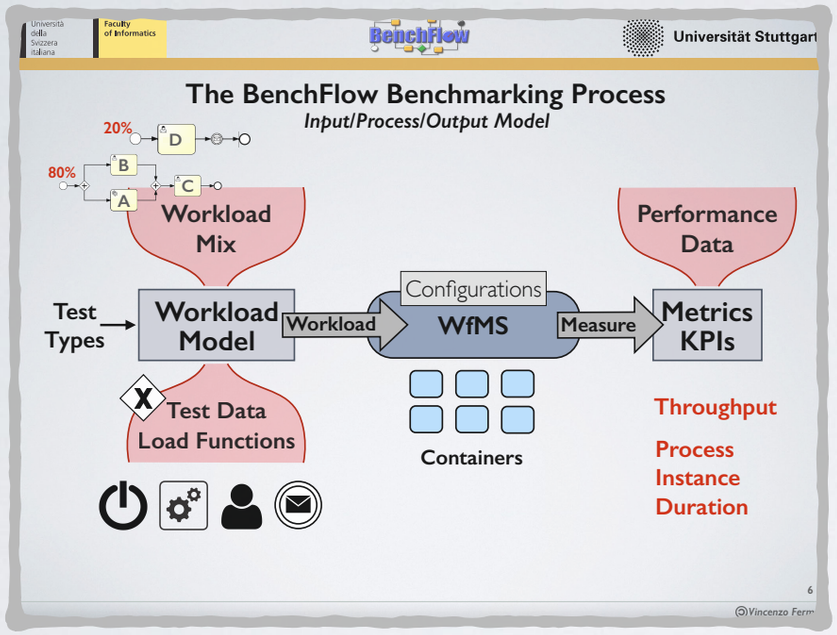
Benchmarking Process



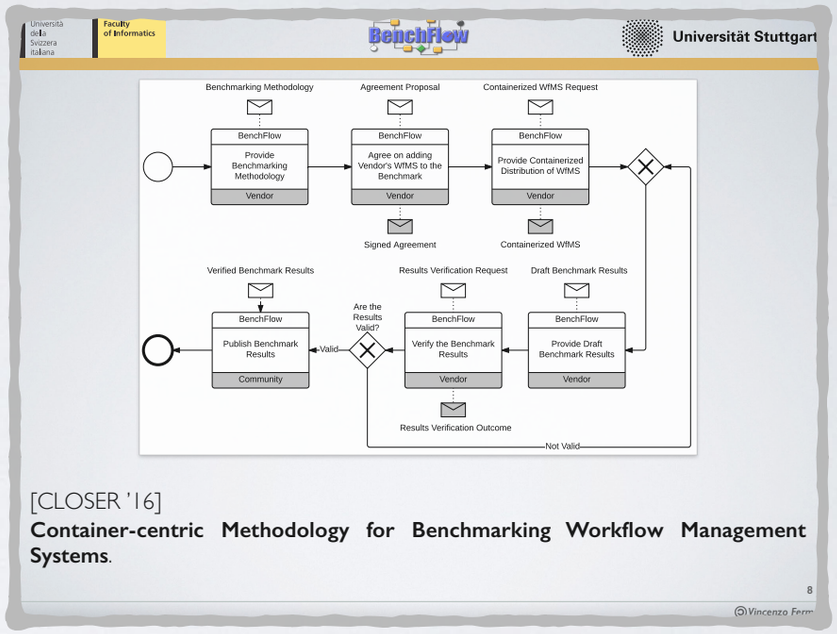
[CLOSER '16]
Container-centric Methodology for Benchmarking Workflow Management Systems.

Benchmarking Methodology

Highlights

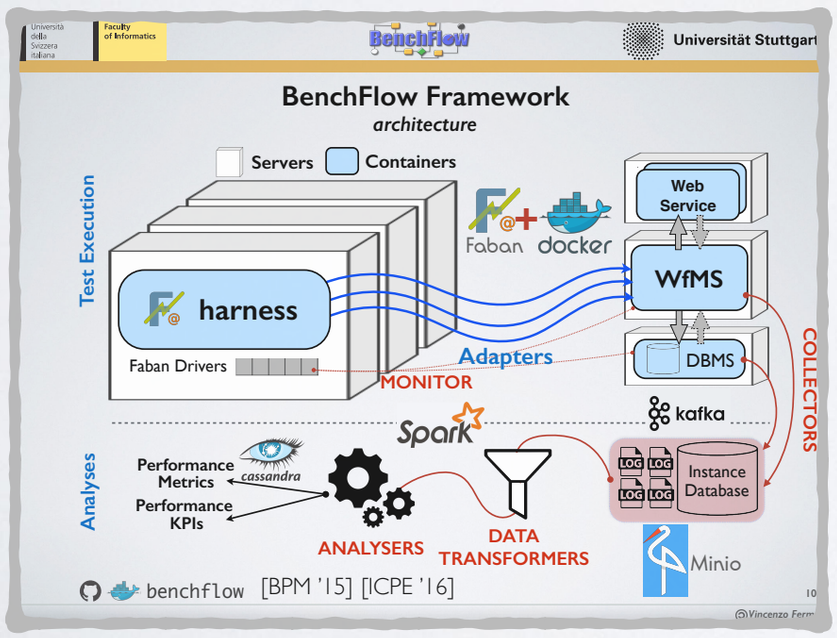


Benchmarking Process



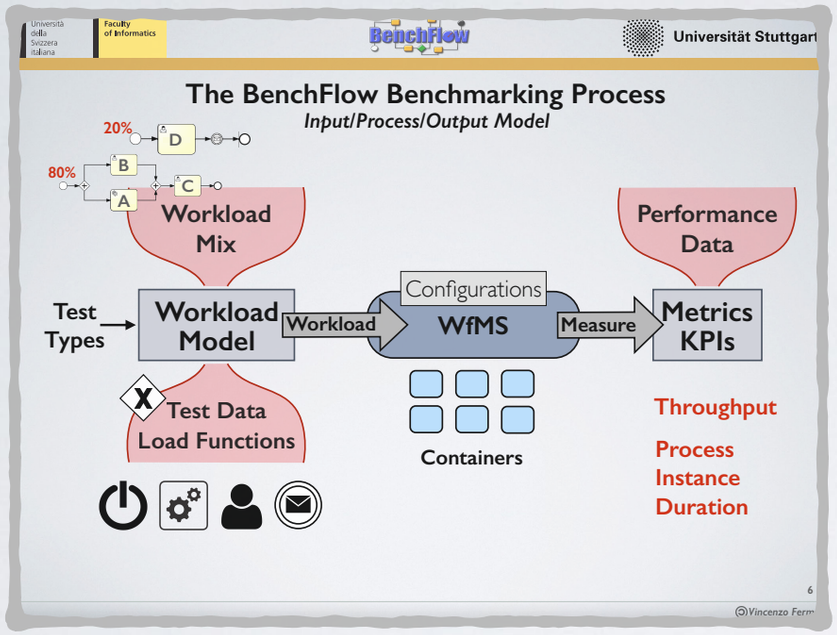
[CLOSER '16]
Container-centric Methodology for Benchmarking Workflow Management Systems.

Benchmarking Methodology

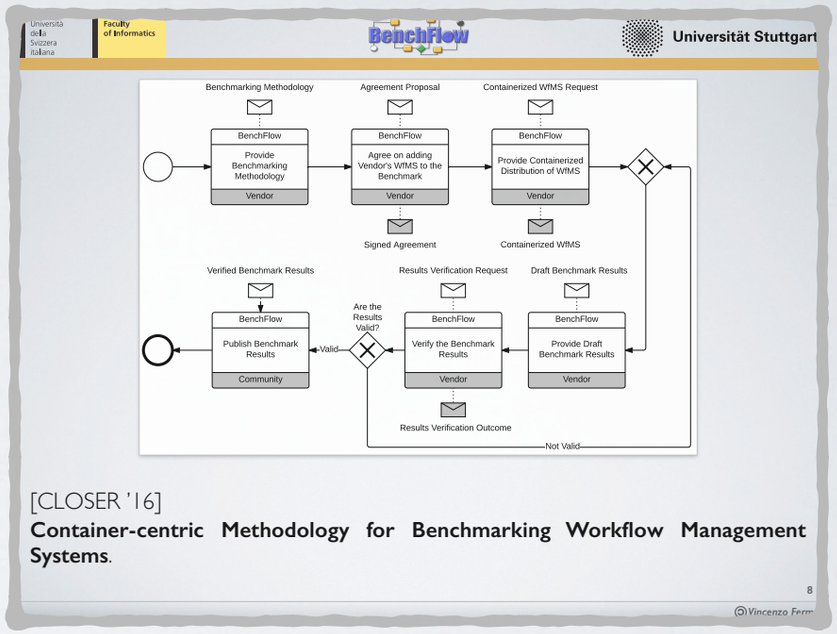


BenchFlow Framework

Highlights

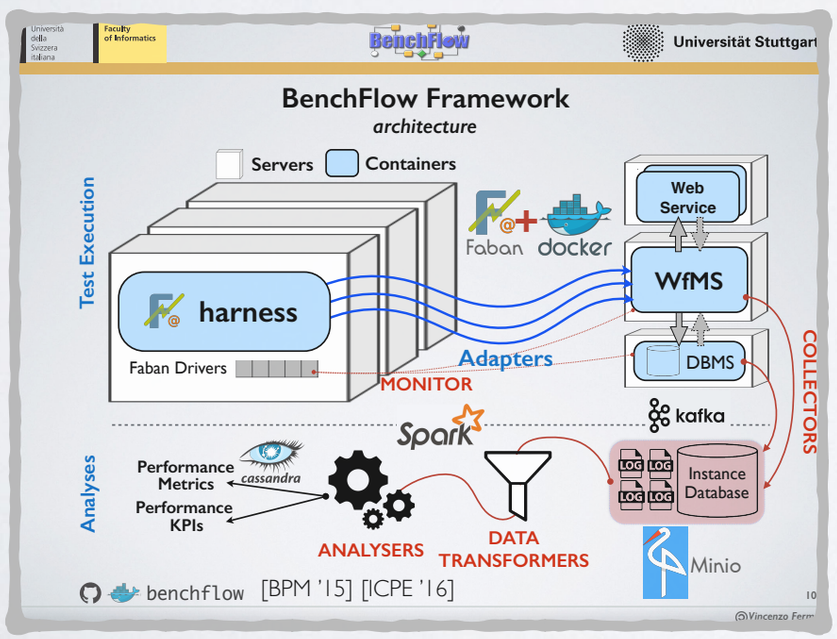


Benchmarking Process

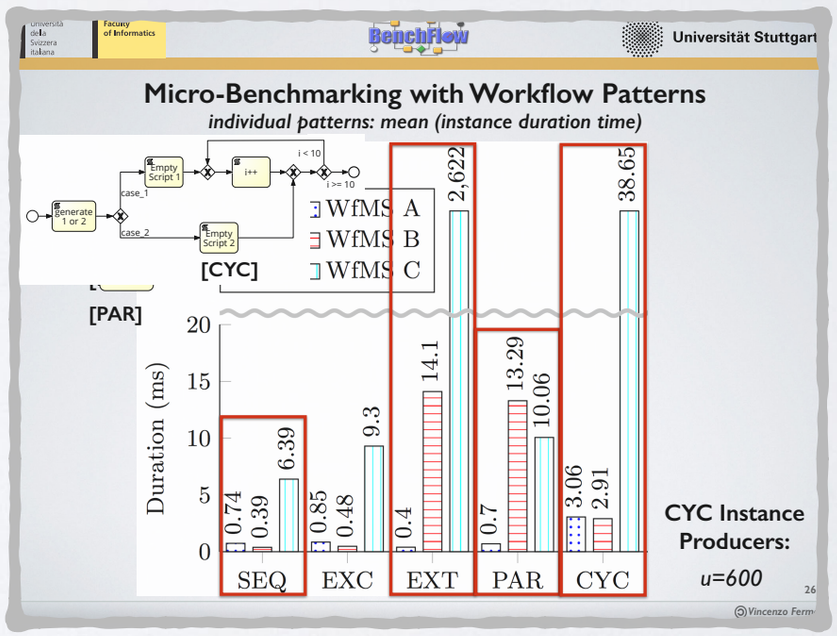


[CLOSER '16]
Container-centric Methodology for Benchmarking Workflow Management Systems.

Benchmarking Methodology



BenchFlow Framework



Micro-Benchmark with WP

Call for Collaboration

WfMSs, process models, process logs

WfMSs

- We want to add more and more WfMSs to the benchmark
- **Contact us for collaboration, and BenchFlow framework support**

Process Models

- We want to characterise the Workload Mix using Real-World process models
- **Share your executable BPMN 2.0 process models, even anonymised**

Execution Logs

- We want to characterise the Load Functions using Real-World behaviours
- **Share your execution logs, even anonymised**

Università della Svizzera italiana
Faculty of Informatics
Universität Stuttgart

WORKFLOW ENGINE PERFORMANCE BENCHMARKING WITH BENCHFLOW

benchflow
benchflow

http://benchflow.inf.usi.ch

✉ vincenzo.ferme@usi.ch

Vincenzo Ferme
Faculty of Informatics
USI Lugano, Switzerland

ESOCC 2016

PEACE
Performance And Conformance Benchmarking Of Workflow Engines

BACKUP SLIDES

Vincenzo Ferme
Faculty of Informatics
USI Lugano, Switzerland

ESOCC 2016

PEACE
Performance And Conformance Benchmarking Of Workflow Engines

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.

[CLOSER '16]
V. Ferme, A. Ivanchikj, C. Pautasso., M. Skouradaki, F. Leymann. **A Container-centric Methodology for Benchmarking Workflow Management Systems**. In Proc. of the 6th International Conference on Cloud Computing and Service Science, CLOSER 2016. SciTePress, 2016.

Published Work

[ICWE '16]
C. Jürgen, V. Ferme, H.C. Gall. **Using Docker Containers to Improve Reproducibility in Software and Web Engineering Research**. In Proc. of the 16th International Conference on Web Engineering, 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.

[ICWS '16]
M. Skouradaki, V. Andrikopoulos, F. Leymann. **Representative BPMN 2.0 Process Model Generation from Recurring Structures**. In Proc. of the 23rd IEEE International Conference on Web Services, ICWS '16, 2016.

Published Work

[SummerSOC '16]
M. Skouradaki, T. Azad, U. Breitenbücher, O. Kopp, F. Leymann. **A Decision Support System for the Performance Benchmarking of Workflow Management Systems**. In Proc. of the 10th Symposium and Summer School On Service-Oriented Computing, SummerSOC '16, 2016.

[BPM Forum '16]
V. Ferme, A. Ivanchikj, C. Pautasso. **Estimating the Cost for Executing Business Processes in the Cloud**. In Proc. of the 14th International Conference on Business Process Management, BPM Forum '16, 2016. (to appear)

[OTM '16]
M. Skouradaki, V. Andrikopoulos, O. Kopp, F. Leymann. **RoSE: Reoccurring Structures Detection in BPMN 2.0 Process Models Collections.** In Proc. of On the Move to Meaningful Internet Systems Conference, OTM '16, 2016. (to appear)

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

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

Configurations

WfMS

BenchFlow Benchmarking Methodology

requirements from the WfMS

CORE

NON-CORE

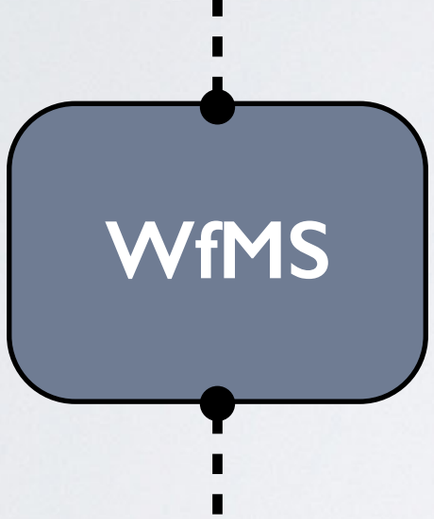
Configurations
WfMS

BenchFlow Benchmarking Methodology

requirements from the WfMS

Initialisation APIs

Deploy Process



Start Process Instance

CORE

NON-CORE

Vincenzo Ferme

Configurations

WfMS

BenchFlow Benchmarking Methodology

requirements from the WfMS

Initialisation APIs

Deploy Process

WfMS

Start Process Instance

User APIs

Create User

Create Group

Pending User Tasks

WfMS

Claim Task

Complete Task

CORE

NON-CORE

Configurations

WfMS

BenchFlow Benchmarking Methodology

requirements from the WfMS

Initialisation APIs

Deploy Process

Start Process Instance

User APIs Web Service APIs

Create User

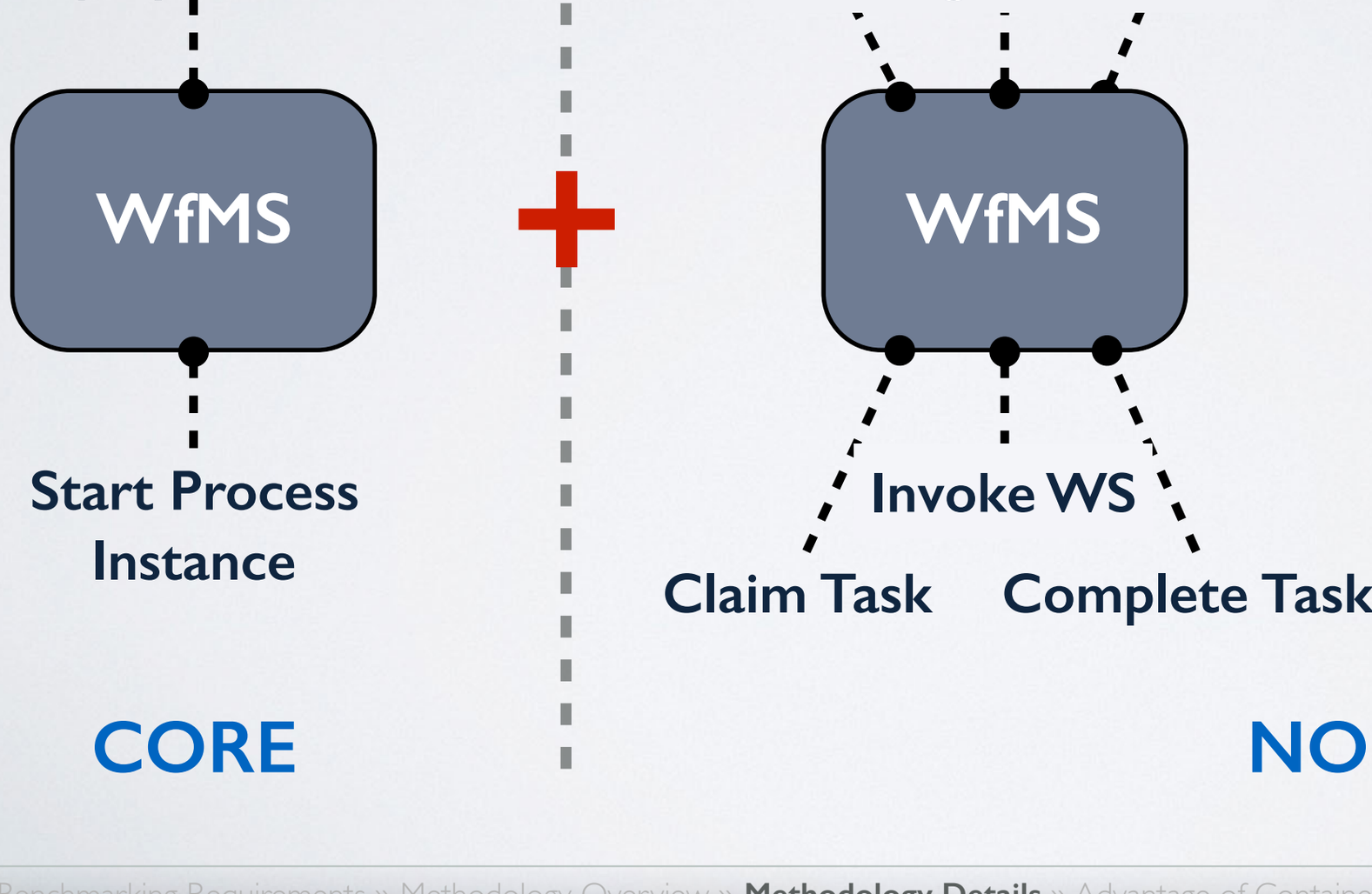
Create Group

Pending User Tasks

Invoke WS

Claim Task

Complete Task



CORE

NON-CORE

Configurations

WfMS

BenchFlow Benchmarking Methodology

requirements from the WfMS

Initialisation APIs

Deploy Process

WfMS

Start Process Instance

User APIs

Create User

Create Group

Pending User Tasks

WfMS

Claim Task

Invoke WS

Complete Task

Web Service APIs

Event APIs

Pending Event Tasks

WfMS

Issue Event

CORE

NON-CORE

Configurations
WfMS

BenchFlow Benchmarking Methodology

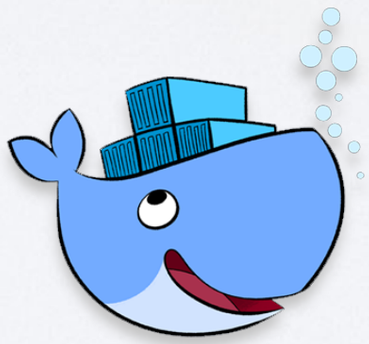
requirements from the WfMS

		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

BenchFlow Framework

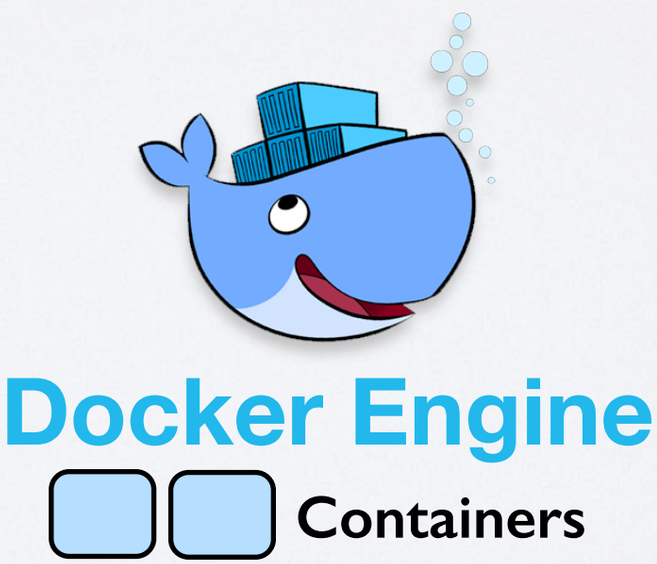
system under test



Docker Engine

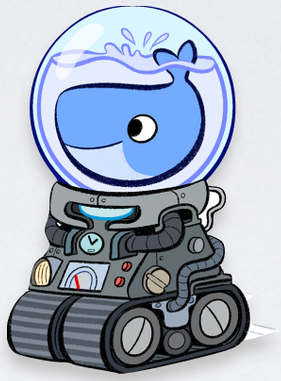
BenchFlow Framework

system under test



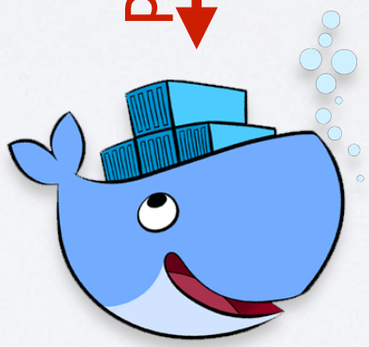
BenchFlow Framework

system under test



Docker Machine

provides

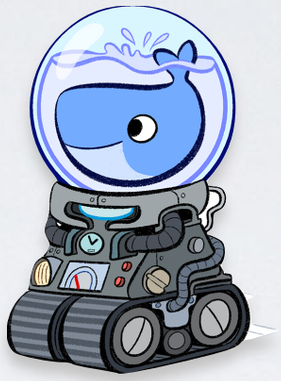


Docker Engine

Containers

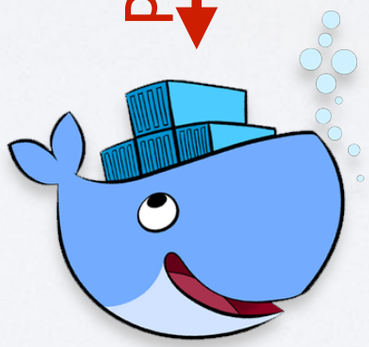
BenchFlow Framework

system under test



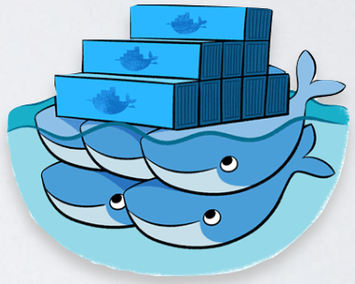
Docker Machine

provides



Docker Engine

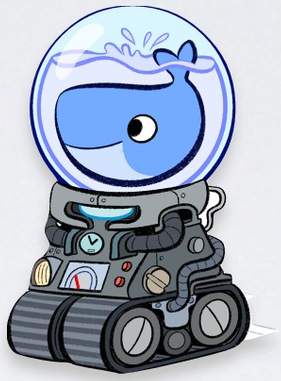
Containers



Docker Swarm

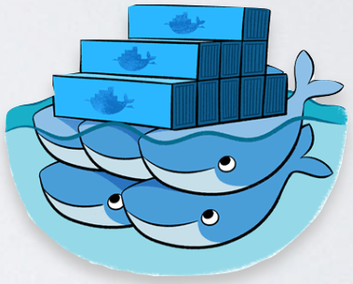
BenchFlow Framework

system under test



Docker Machine

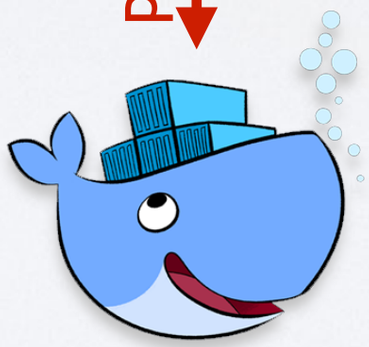
provides



Docker Swarm

manages

Servers

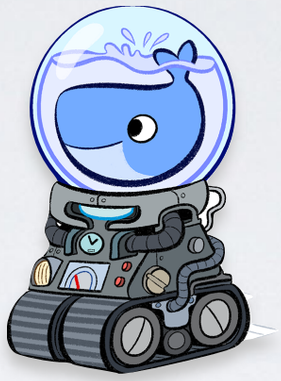


Docker Engine

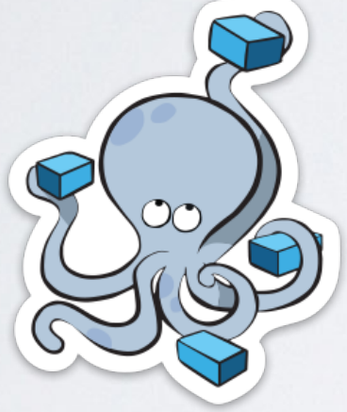
Containers

BenchFlow Framework

system under test

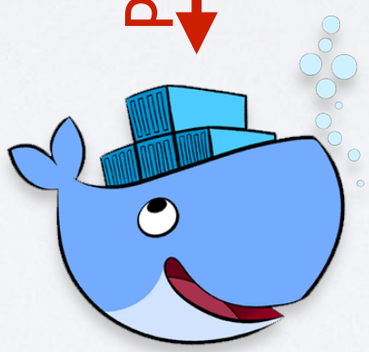


Docker Machine



Docker Compose

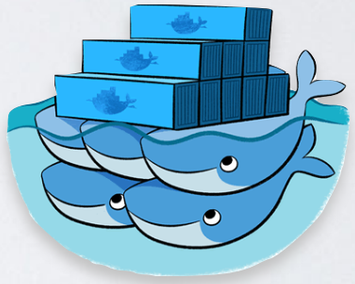
provides



Docker Engine

Containers

manages

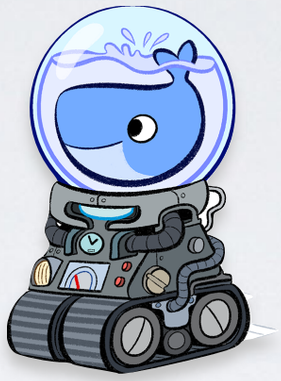


Docker Swarm

Servers

BenchFlow Framework

system under test



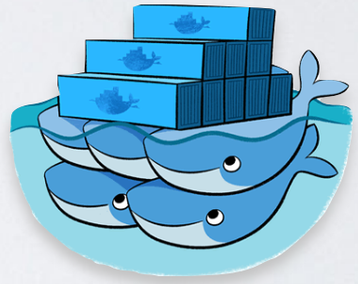
Docker Machine

Docker Compose

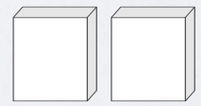
SUT's Deployment Conf.

deploys

provides

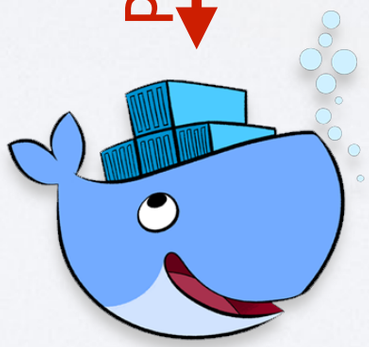


Docker Swarm



Servers

manages



Docker Engine

Containers

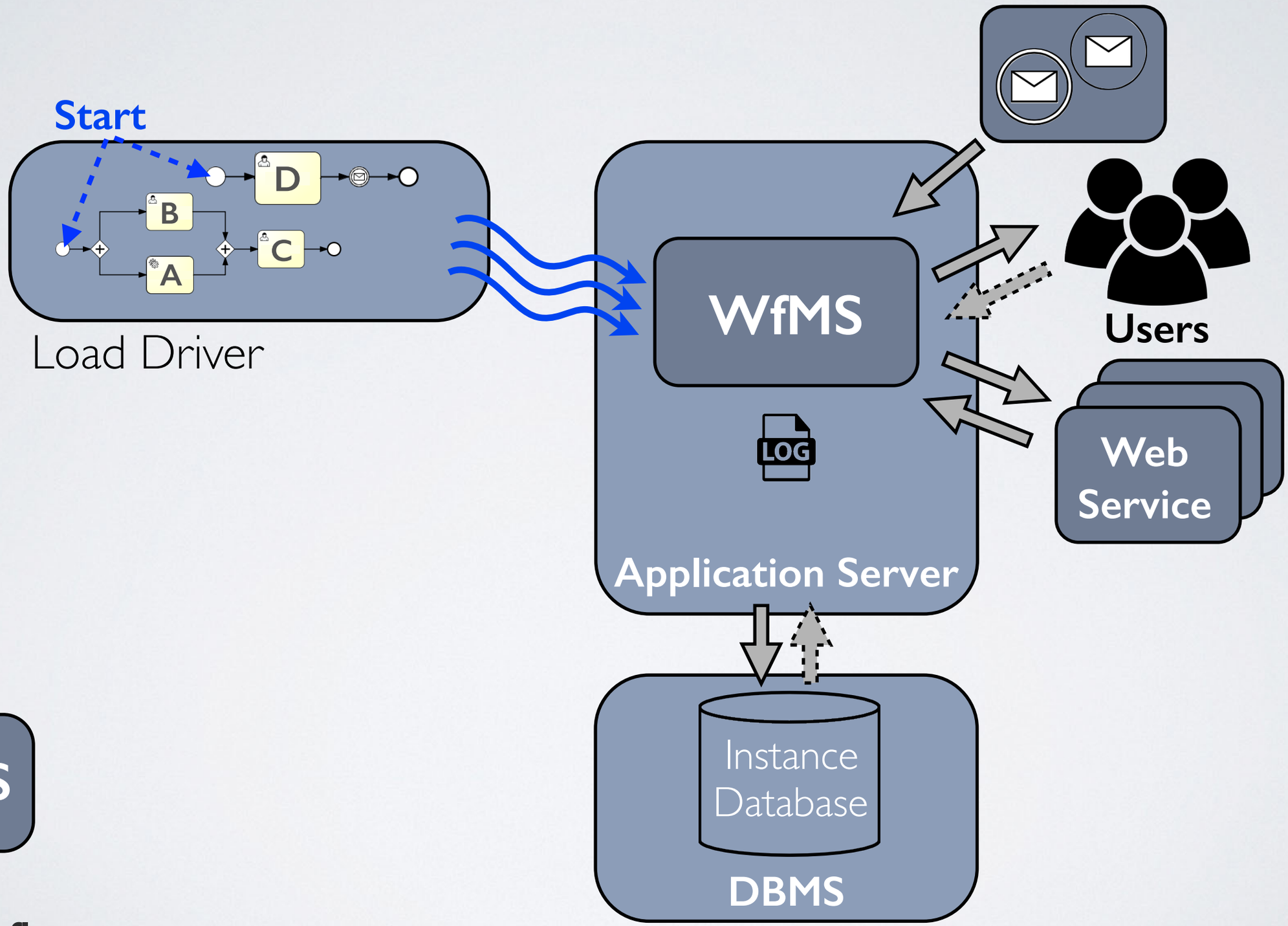
Server-side Data and Metrics Collection

asynchronous execution of workflows

Start Workflow

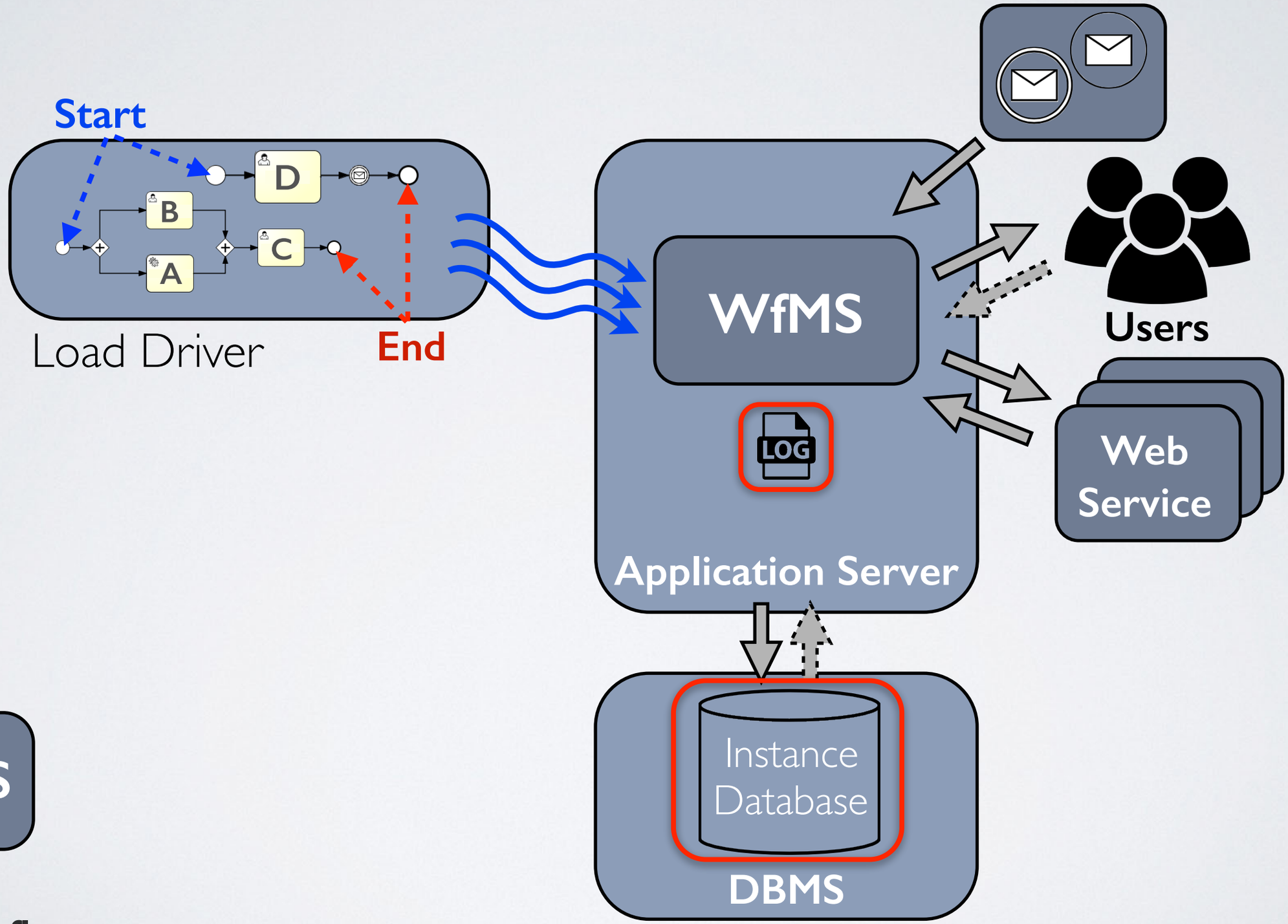
Server-side Data and Metrics Collection

asynchronous execution of workflows



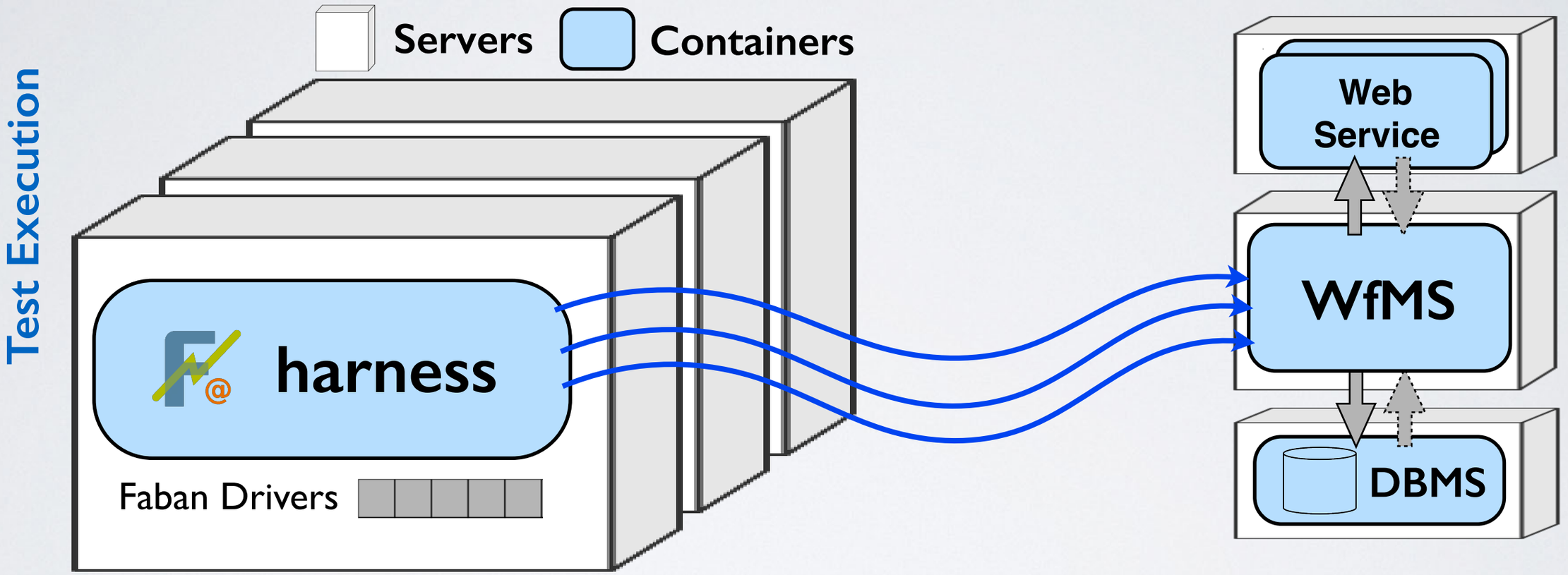
Server-side Data and Metrics Collection

asynchronous execution of workflows



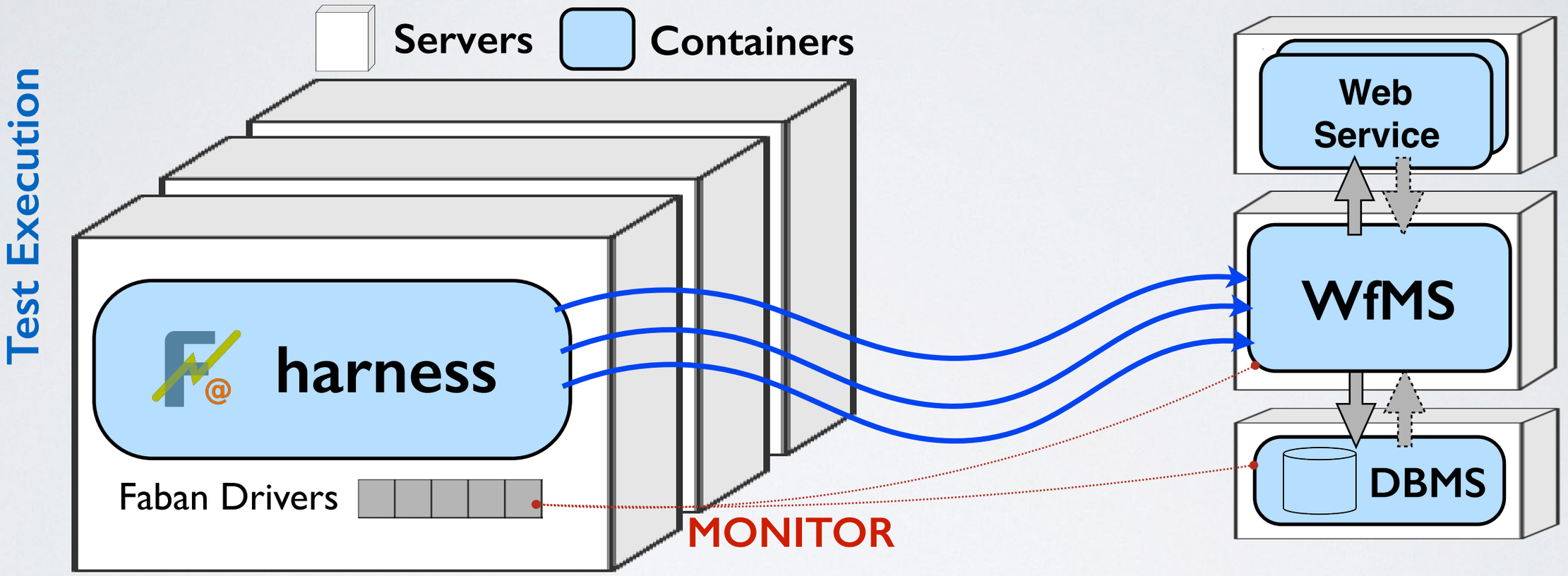
Server-side Data and Metrics Collection

monitors



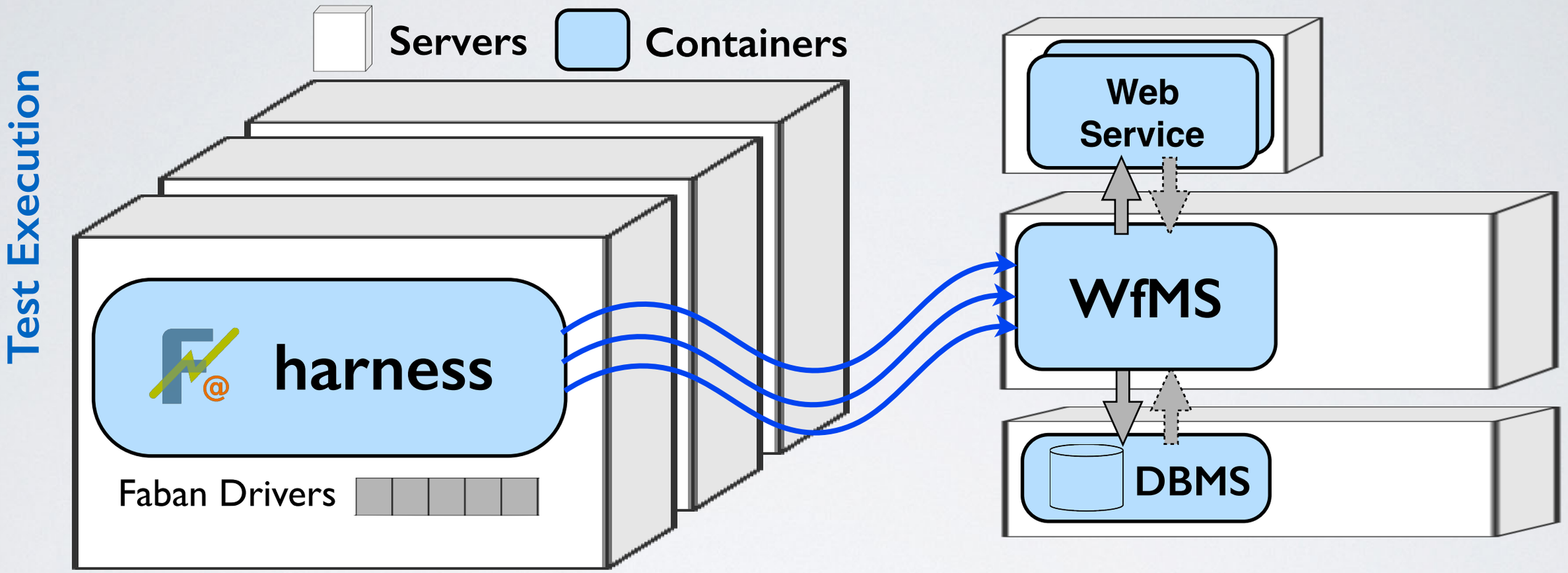
Server-side Data and Metrics Collection

monitors



Server-side Data and Metrics Collection

monitors



Monitors' Characteristics:

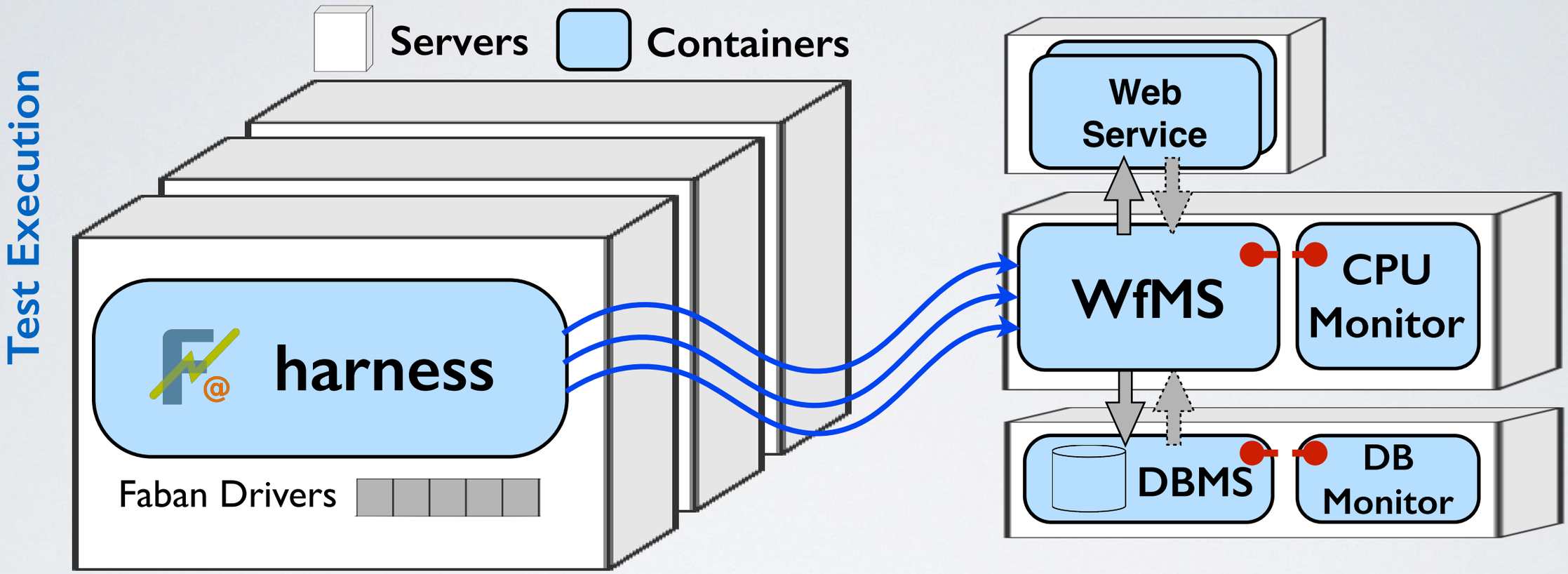
- RESTful services
- Lightweight (written in Go)
- As less invasive on the SUT as possible

Examples of Monitors:

- CPU usage
- Database state

Server-side Data and Metrics Collection

monitors



Monitors' Characteristics:

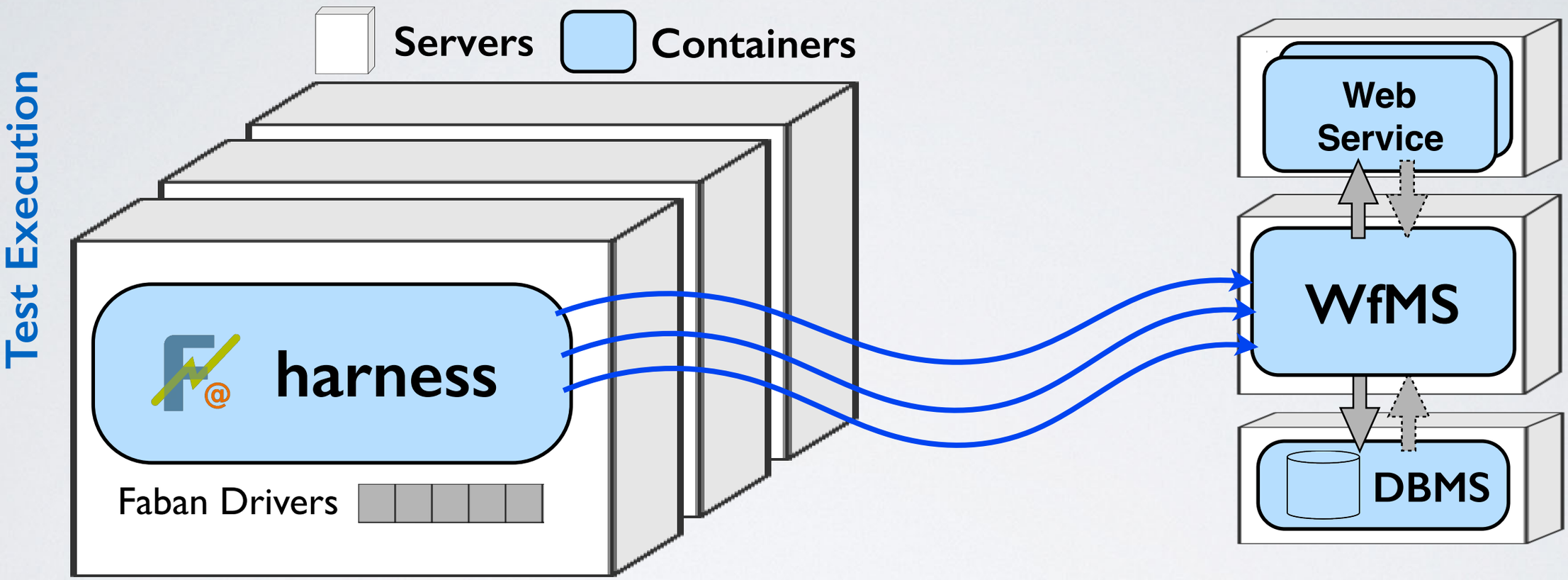
- RESTful services
- Lightweight (written in Go)
- As less invasive on the SUT as possible

Examples of Monitors:

- CPU usage
- Database state

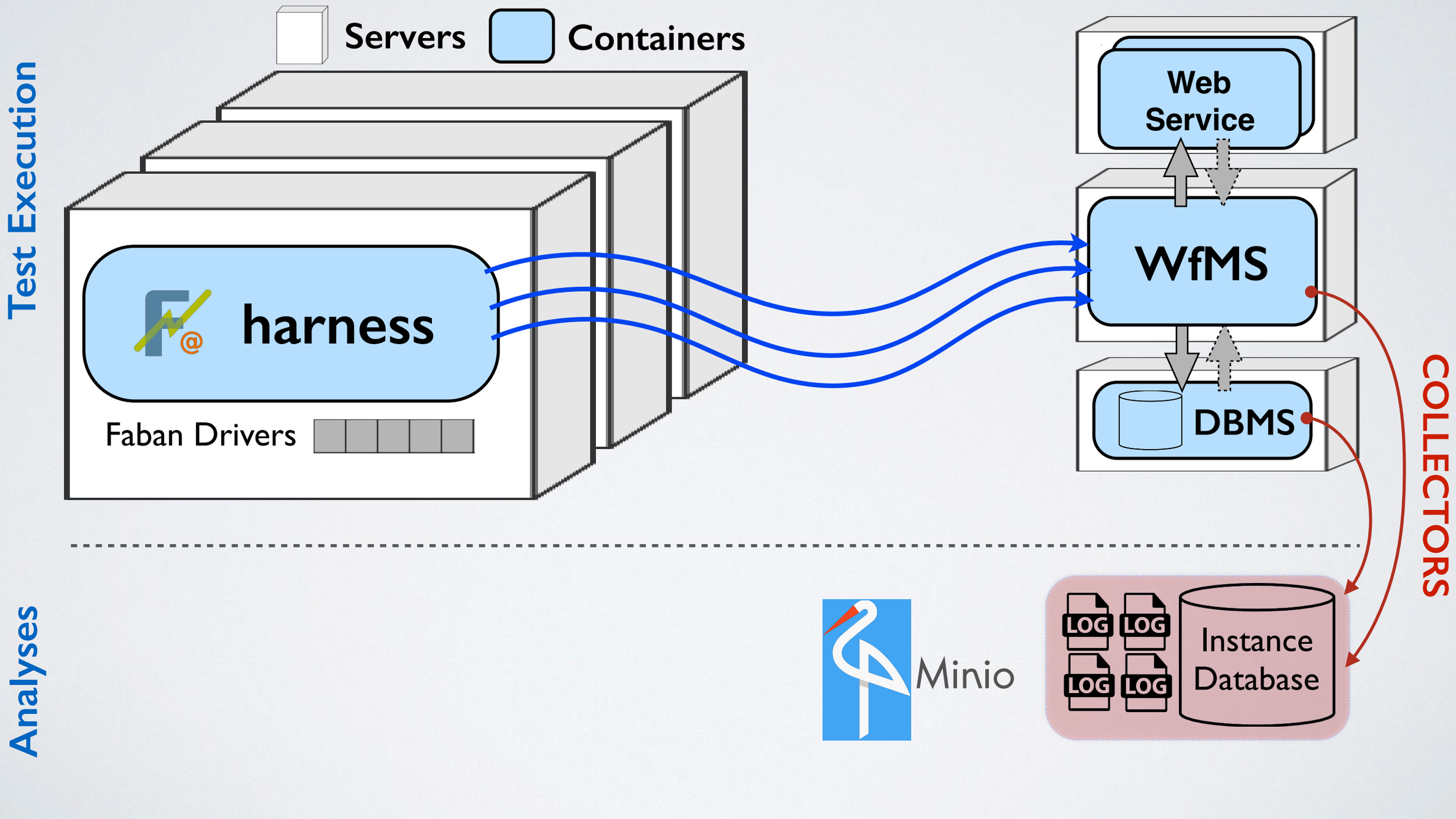
Server-side Data and Metrics Collection

collect data



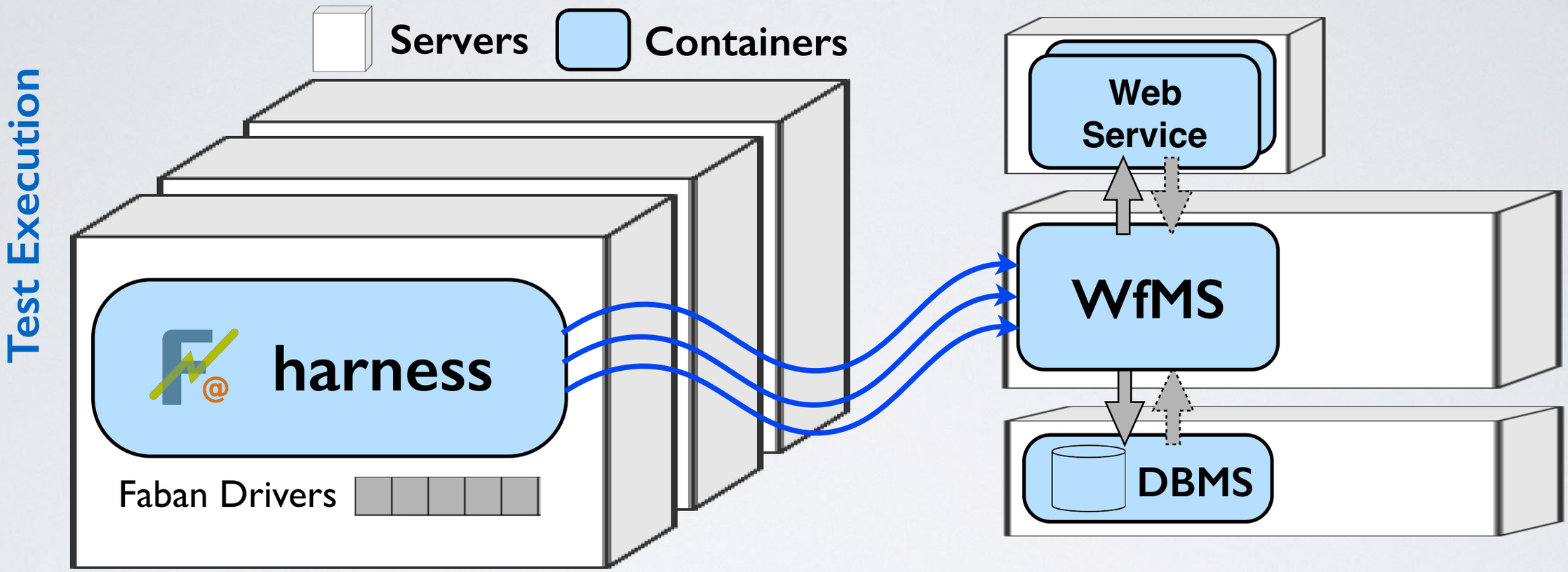
Server-side Data and Metrics Collection

collect data



Server-side Data and Metrics Collection

collect data



Collectors' Characteristics:

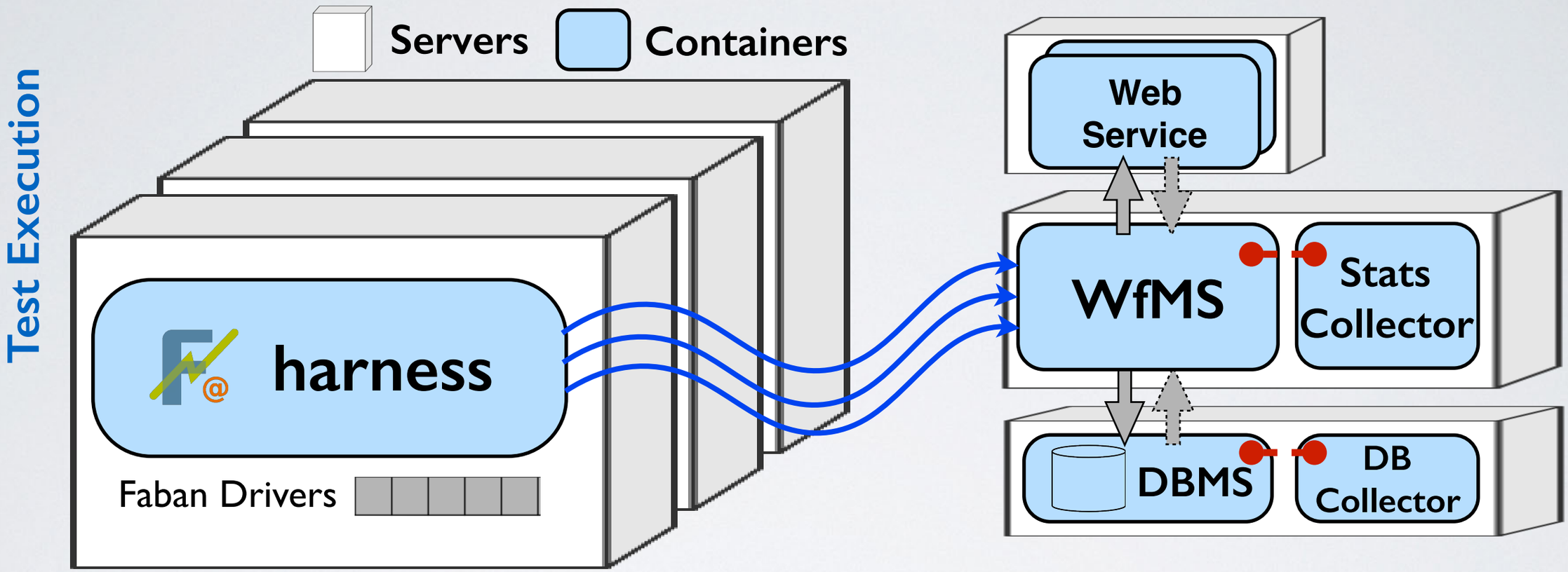
- RESTful services
- Lightweight (written in Go)
- Two types: online and offline
- Buffer data locally

Examples of Collectors:

- Container's Stats (e.g., CPU usage)
- Database dump
- Applications Logs

Server-side Data and Metrics Collection

collect data



Collectors' Characteristics:

- RESTful services
- Lightweight (written in Go)
- Two types: online and offline
- Buffer data locally

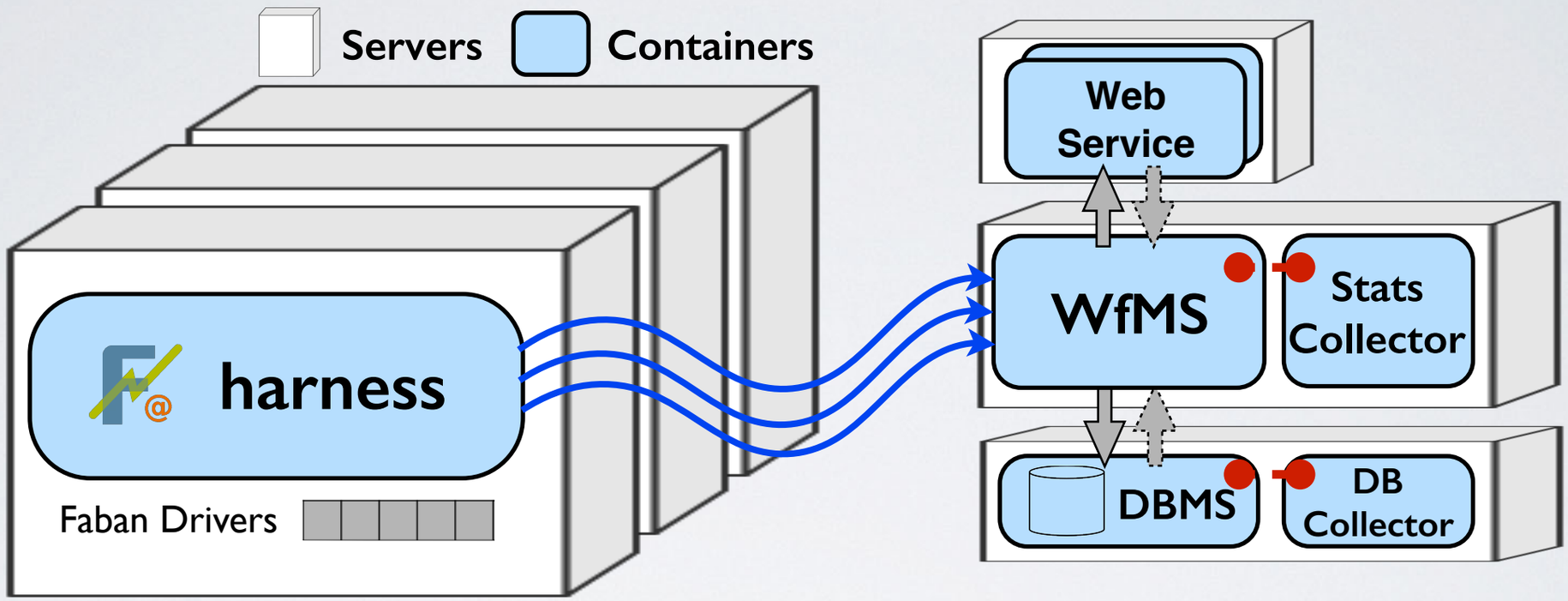
Examples of Collectors:

- Container's Stats (e.g., CPU usage)
- Database dump
- Applications Logs

Synchronisation with the Analyses

schedule analyses

Test Execution

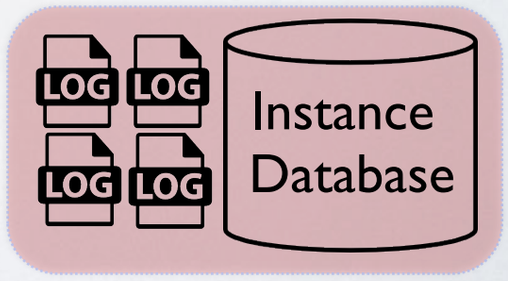


Analyses

kafka — A high-throughput distributed messaging system

Spark

Minio



Synchronisation with the Analyses

schedule analyses

Test Execution

Servers

Containers

harness

Faban Drivers

Web Service

WfMS

Stats Collector

DBMS

DB Collector

COLLECT

kafka

A high-throughput distributed messaging system

Analyses

Spark

Minio

LOG LOG LOG LOG

Instance Database

Synchronisation with the Analyses

schedule analyses

Test Execution

Servers

Containers

harness

Faban Drivers

Web Service

WfMS

Stats Collector

DBMS

DB Collector

Publish

COLLECT

kafka

A high-throughput distributed messaging system

Analyses

Spark

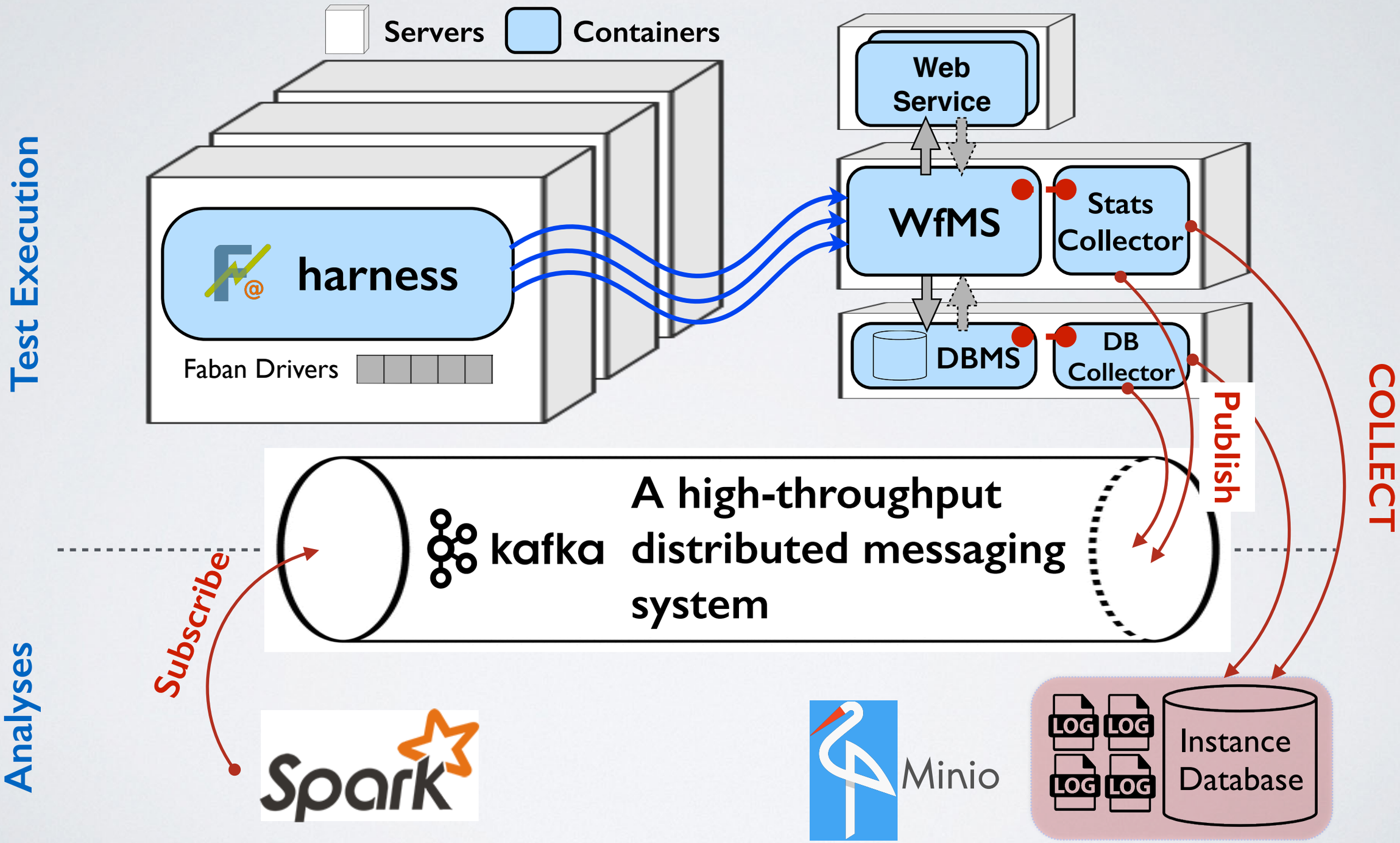
Minio

LOG LOG LOG LOG

Instance Database

Synchronisation with the Analyses

schedule analyses



Synchronisation with the Analyses

schedule analyses

Test Execution

Servers

Containers

harness

Faban Drivers

Web Service

WfMS

Stats Collector

DBMS

DB Collector

Publish

COLLECT

kafka

A high-throughput distributed messaging system

Subscribe

Analyses

Spark

Read

Minio

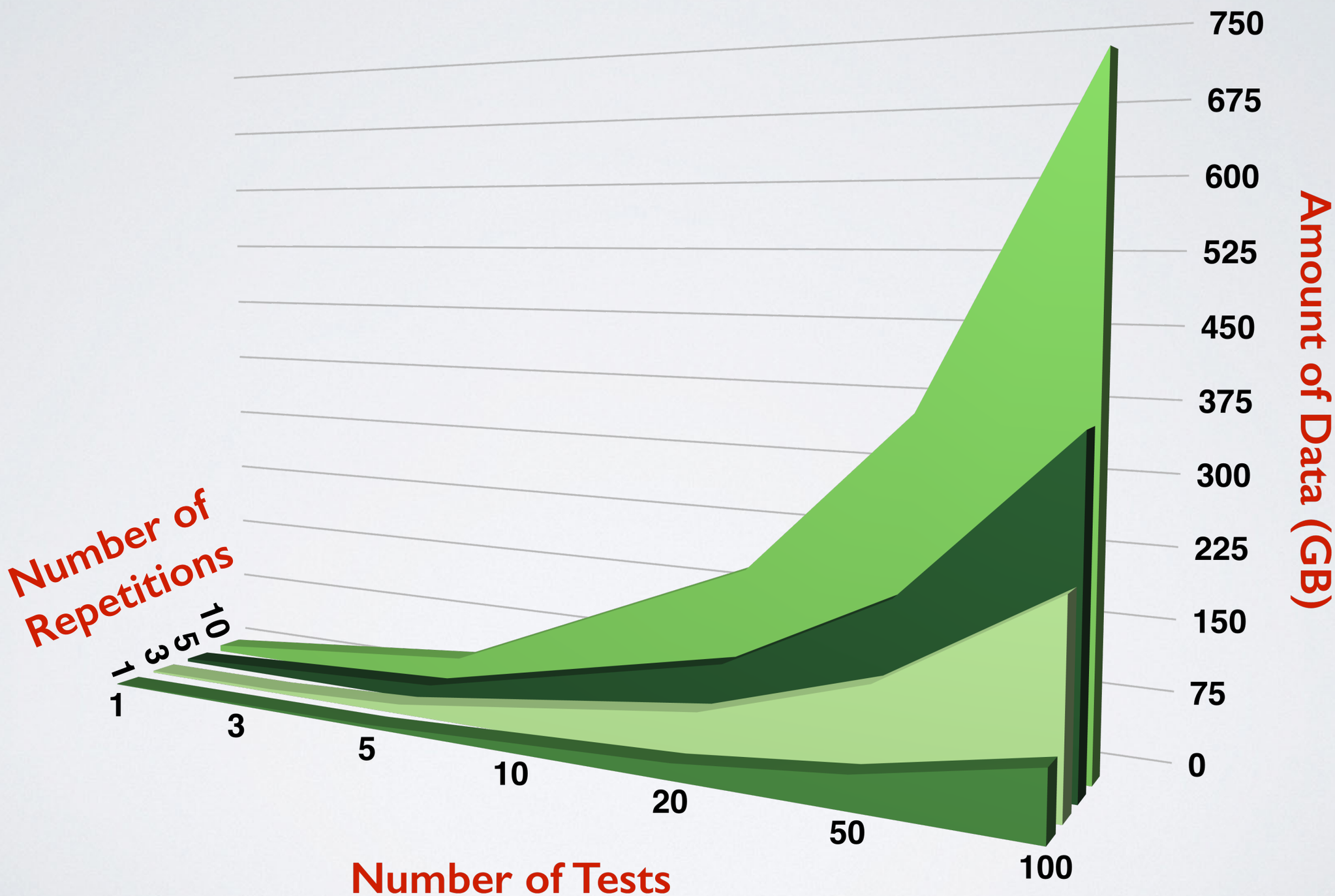
LOG LOG LOG LOG

Instance Database

REALISTIC DATA

Performance Metrics and KPIs

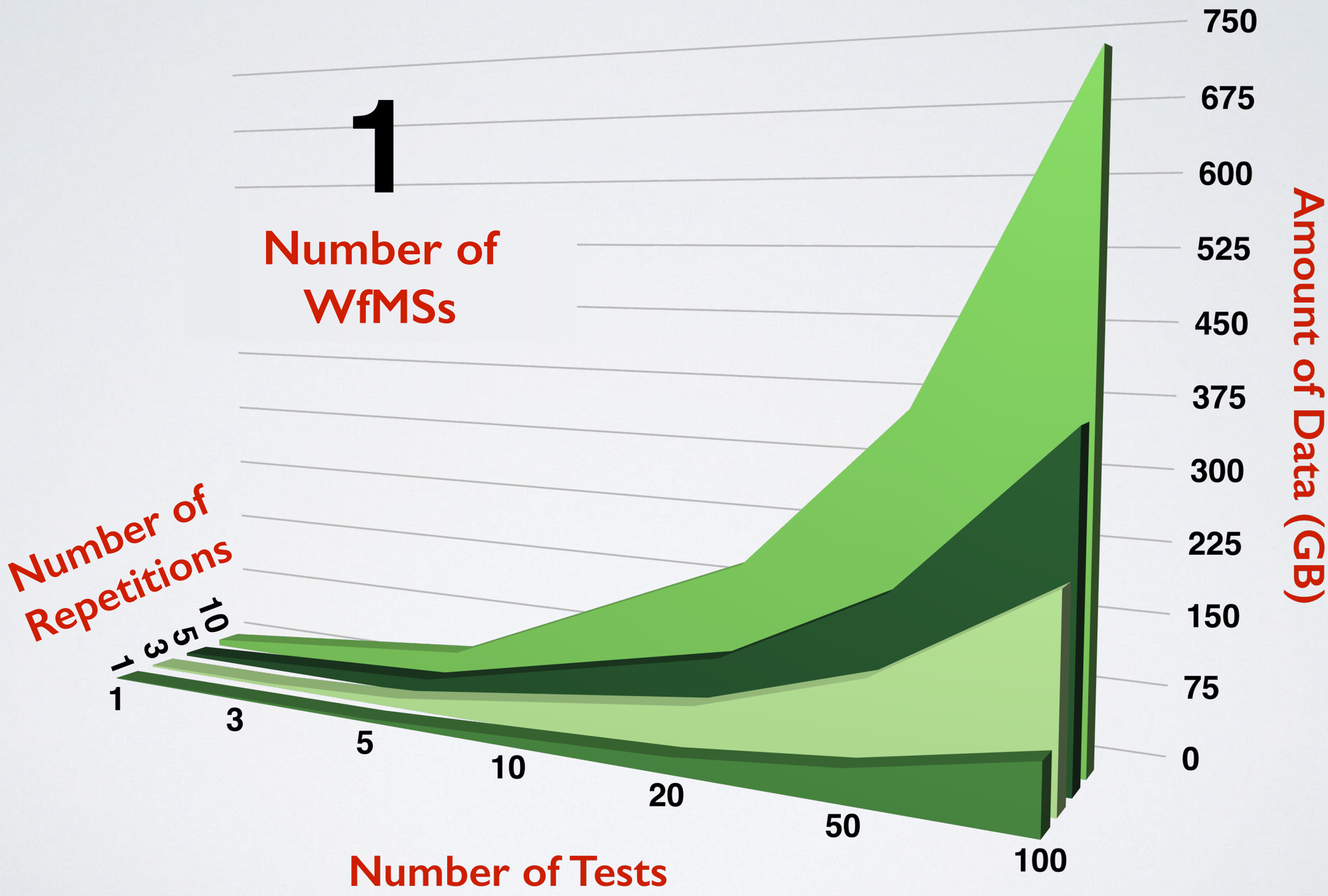
amount of data



REALISTIC DATA

Performance Metrics and KPIs

amount of data



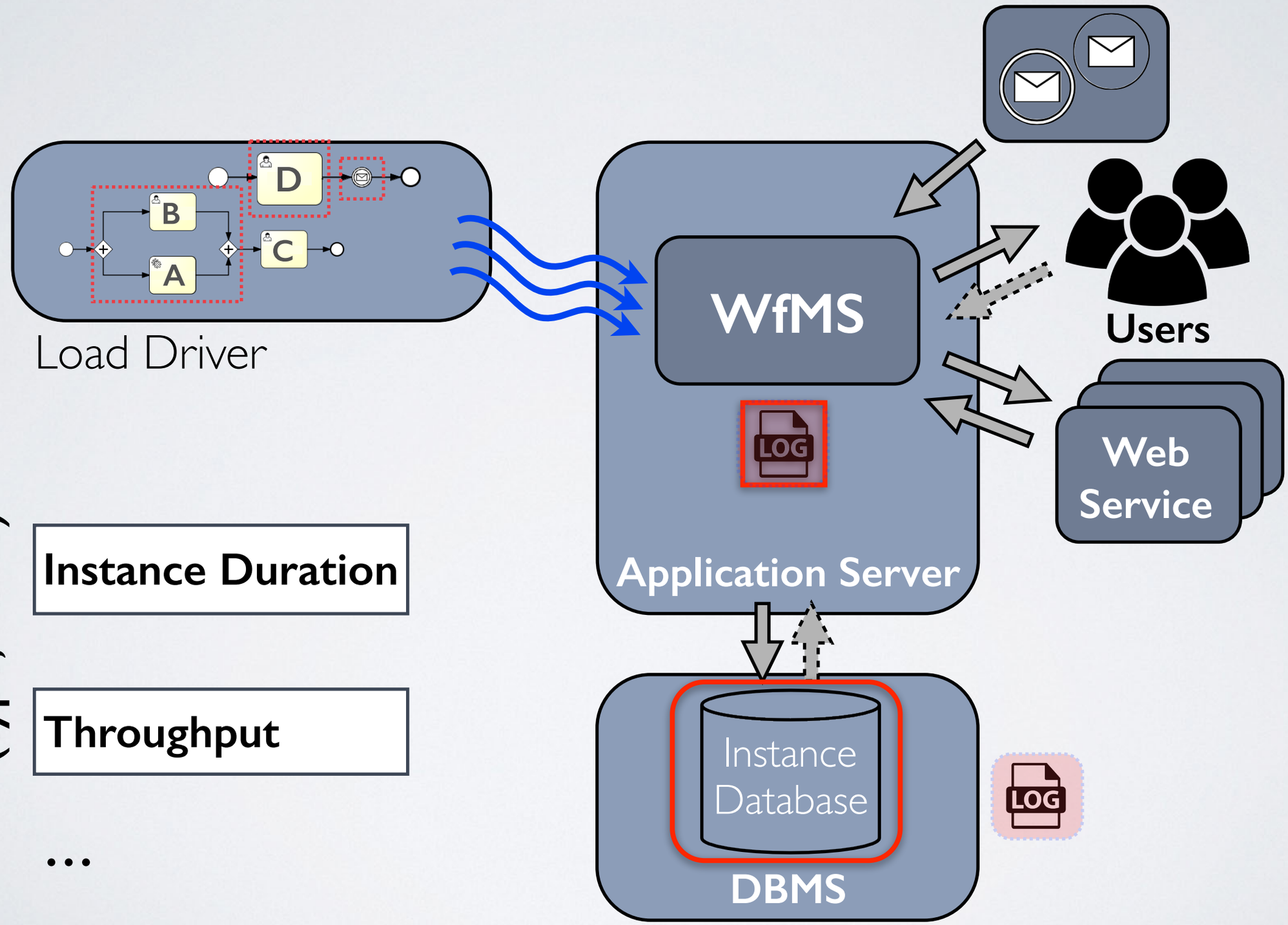
Micro-Benchmarking with Workflow Patterns

mix of patterns: throughput (bp/sec)

MIX	
WfMS A	1061.27
WfMS B	1402.33
WfMS C	1.78

Computed Metrics and Statistics

feature level metrics



Per Construct (type, name)

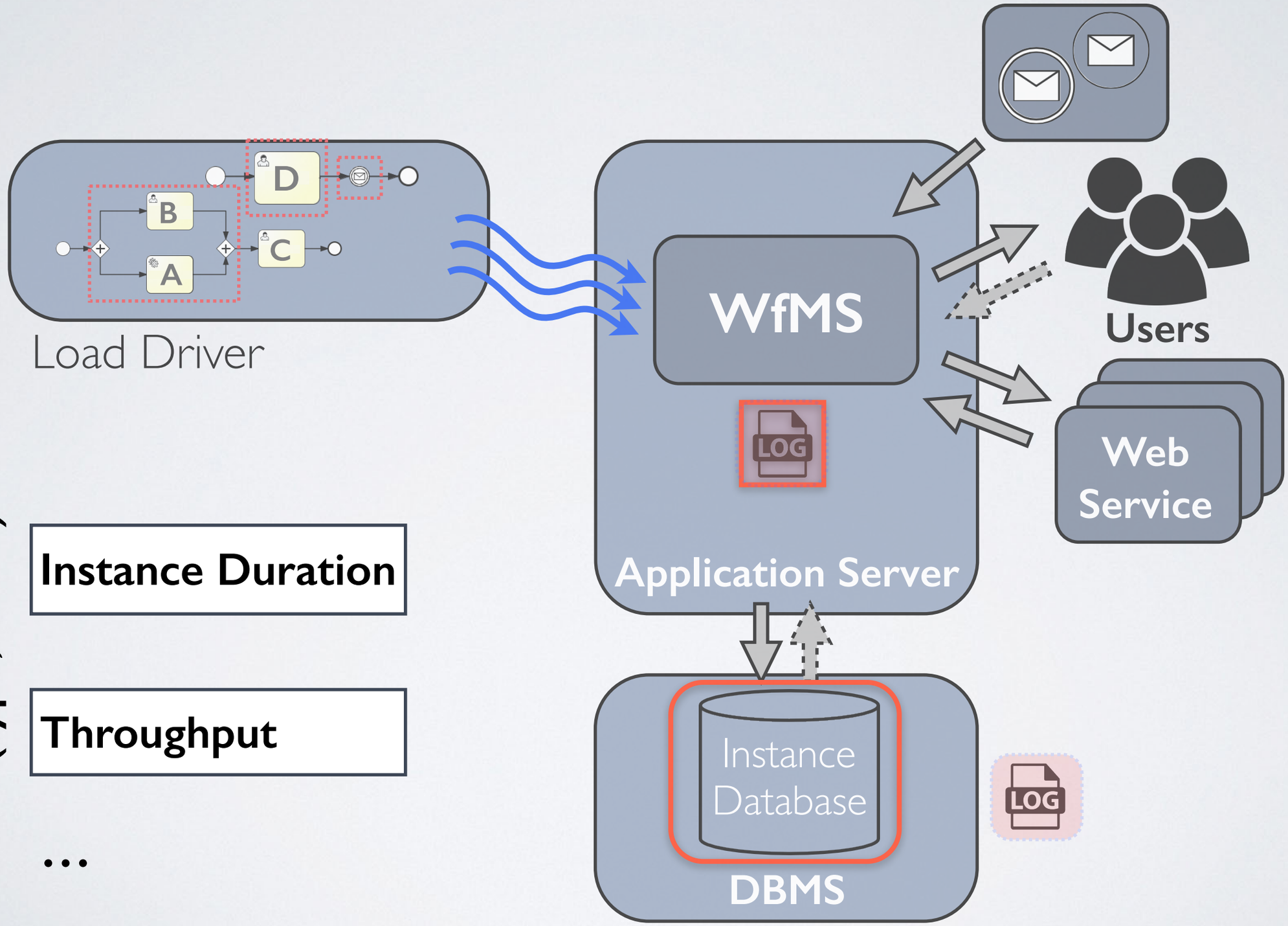
- **Instance Duration**
- **Throughput**
- …

Computed Metrics and Statistics

feature level metrics

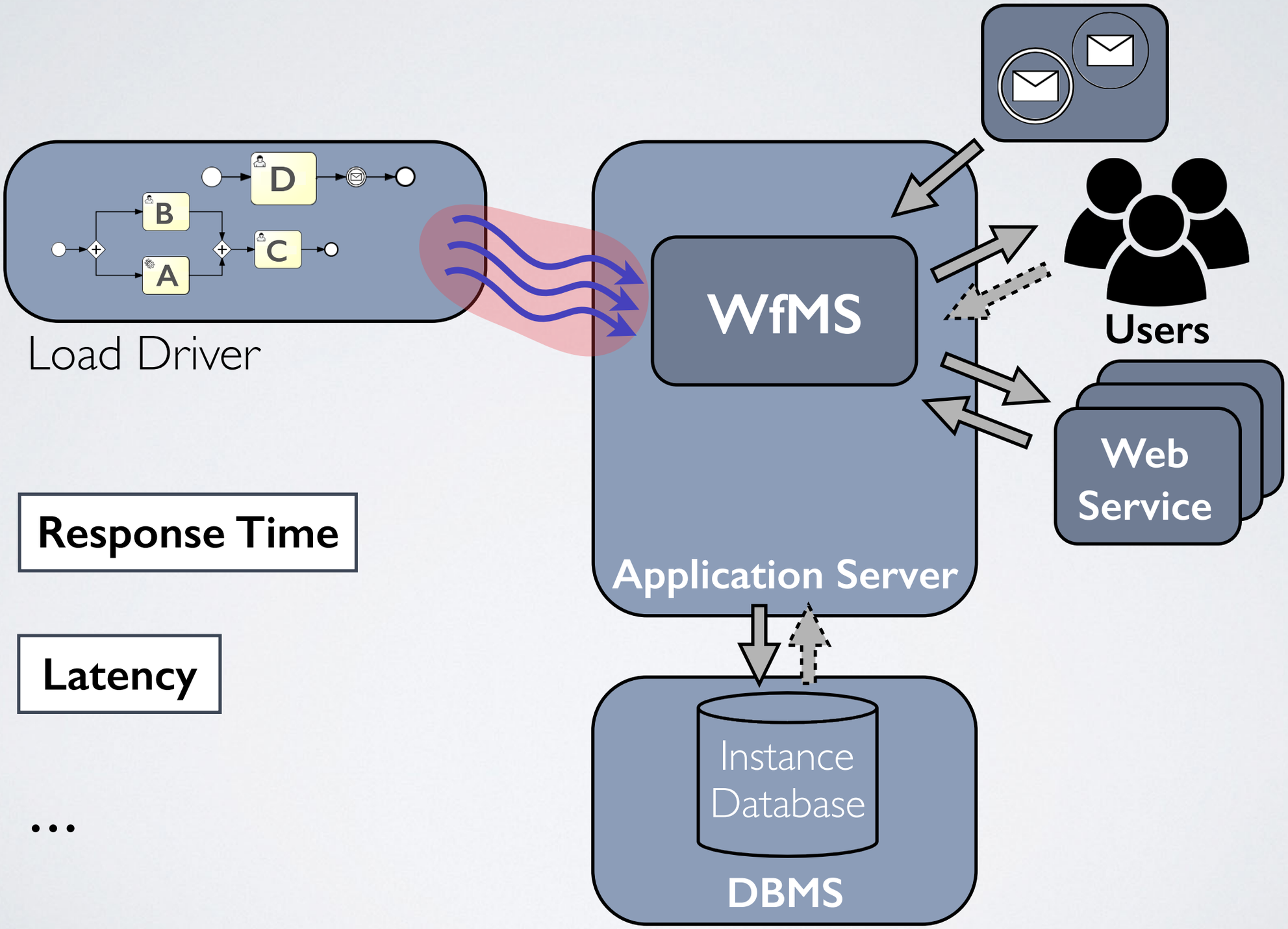
Per Construct (type, name)

- **Instance Duration**
- **Throughput**
- …



Computed Metrics and Statistics

interactions metrics



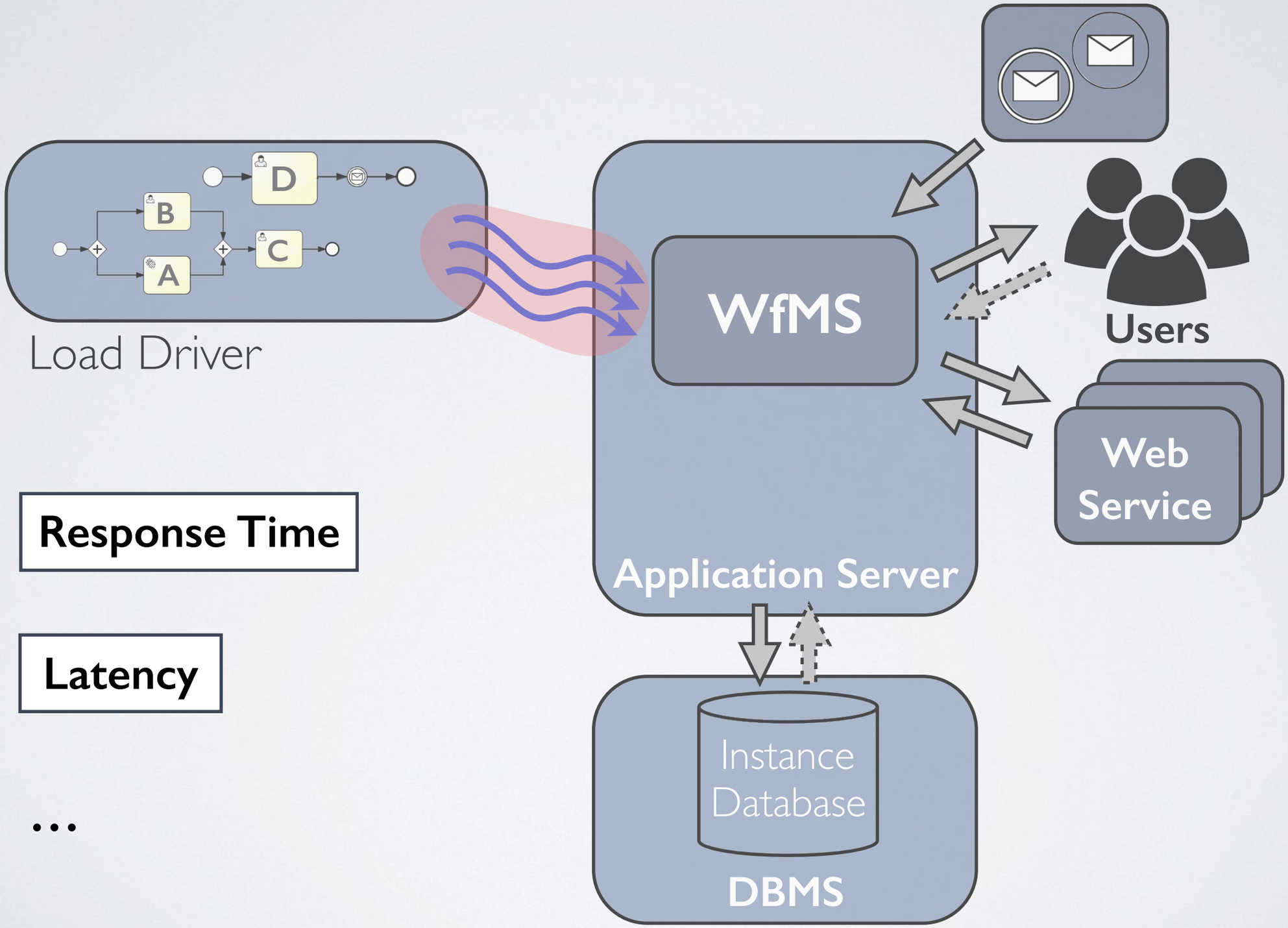
Response Time

Latency

...

Computed Metrics and Statistics

interactions metrics



Response Time

Latency

…

Computed Metrics and Statistics

descriptive and homogeneity statistics

- Descriptive Statistics (e.g., Mean, Confidence Interval, Standard Deviation, Percentiles, Quartiles, …)
- Coefficient of Variation across trials
- Levene Test on trials