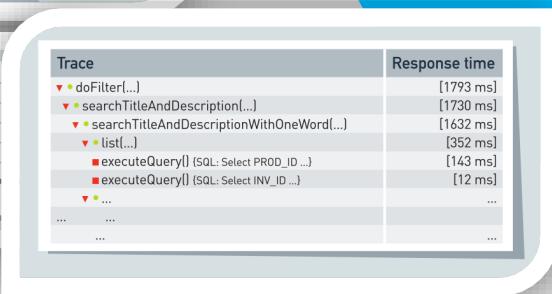
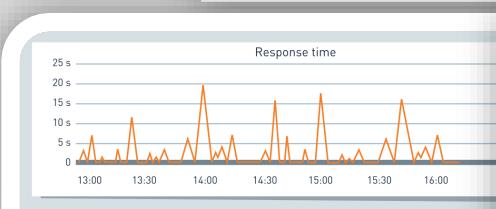


State of the Art of Visualization in APM Tools

André van Hoorn
Dušan Okanović



State of the Art of Visualization in APM Tools

Part 1/2: Introduction to APM

André van Hoorn
Dušan Okanović

[Alle Events](#)[Alle Orte](#)

Suche nach Künstlern und Events

Suchen

Warenkorb

Konzerte

Kultur

Sport

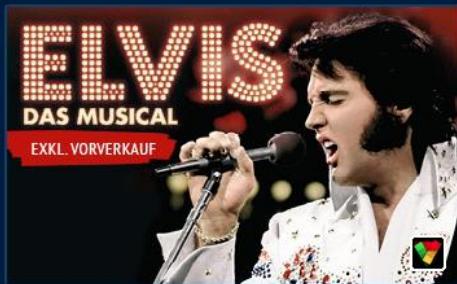
Musical & Show

Humor

Eventreisen

Mehr...

Special Tickets



EVENTIM Charts



Andreas Gabalier
★★★★★
Tickets ab € 49,90

02

Mark Forster

Tipps (16)



OTTO

Übersicht

Übersicht

Tickets

Fan-Reports

Biografie

Gefällt mir

Teilen



Vorhang auf: OTTO bei eventim.de

OTTO – der Name ist Programm: Ostfriesisch Temperamentvoll Total Onverwüstlich. „Otto Live“ ist weiter auf Tour. Die Einschätzung einiger Kritiker, Otto wäre der beste Bühnenkomiker seiner Generation, ist natürlich barer Blödsinn. Wahr ist allerdings, dass es weit und breit niemanden gibt, der ihm auf der Bühne das Wasser reichen kann – schon weil er wie immer ganz allein auftreten muss. Andere werfen Otto vor, dass er aus seinem Talent bisher zu wenig gemacht habe – auch die dürfen beruhigt sein: Er arbeitet ja weiter daran.

Ticketalarm – kein Event mehr verpassen!

Registrieren Sie sich für den eventim.de-Ticketalarm, und Sie werden per E-Mail informiert, sobald es neue Termine Ihrer Lieblingsstars und -events gibt.

REGISTRIEREN SIE SICH JETZT FÜR OTTO

 Ihre E-Mail-Adresse[Jetzt anmelden! ▶](#)

Ich möchte den Ticketalarm des aufgeführten Künstlers abonnieren. Hierzu darf CTS EVENTIM AG & Co. KGaA meine E-Mail-Adresse verwenden. Die Abmeldung des Ticketalarms ist jederzeit möglich. EVENTIM legt großen Wert auf Datenschutz. Die Datenschutzerklärung können Sie [hier](#) nachlesen.

Über diesen Künstler

Künstler-Biografie

Otto Gerhard Waalkes, häufig einfach nur Otto genannt, (* 22. Juli 1948 in Emden) ist ein deutscher Komiker, Comiczeichner, Musiker, Schauspieler, Regisseur und Synchronsprecher. Der gebürtige Ostfriese gilt als einer der erfolgreichsten Vertreter des deutschen Humors. Im Jahr 2007 kam Otto Waalkes bei der Wahl zum besten deutschsprachigen Komiker ...

[► weiterlesen](#)



Tickets, Konzertkarten & Eintrittskarten

Login

DE

Alle Events

Alle Orte

Suche nach Künstlern und Events

Suchen

Warenkorb



MOMENT, BITTE!

SIE BEFINDEN SICH IM WARTERAUM VON EVENTIM.
BITTE VERSUCHEN SIE ES SPÄTER WIEDER.

“Application performance management (APM), as a core IT operations discipline, aims to achieve an adequate level of performance during operations. To achieve this,

APM comprises **methods, techniques, and tools** for

- **continuously monitoring** the state of an application system and its usage, as well as for
- **detecting, diagnosing, and resolving performance-related problems** using the monitored data.”



Application Performance Management: State of the Art and Challenges for the Future

Christoph Heger¹, André van Hoorn¹, Mario Mann¹, Dusan Okanović²
¹NexTec Consulting GmbH, Competence Area APM, Lenfelder-Eckertengasse, Germany
²University of Stuttgart, Institute of Software Technology Stuttgart, Germany

ABSTRACT

The performance of application systems has a direct impact on business metrics. For example, companies face customers and revenue losses if their web sites are slow or have high response times. Application performance management (APM) aims to support IT operations teams in maintaining a continuous and specialized picture of system performance. This paper provides an overview of the state-of-the-art in APM and evolution of performance-related methods. It highlights the challenges of APM and discusses the state of art in APM in industrial practice and academic research. Finally, it identifies research directions for future work and research directions.



Figure 1: Continuous APM activities

1. INTRODUCTION

Business success is directly influenced by the performance of the enterprise application systems that support it. Any performance degradation of such applications may bring losses in revenue, and even more serious consequences, such as loss of customers and damage to their image are well documented. Google loses 20% traffic if their search engines are slow [1]. In 2012, 40% of users left a website if the page load time was over 3 seconds [2]. Mozilla's study showed that 40% of users leave a website if the page load time is over 3 seconds [3]. In 2013, 50% of users will leave the web site if they are not satisfied with the response time [4]. Application performance management (APM), as a core IT operations discipline, aims to achieve an adequate level of performance of application systems using APM-specific methods, techniques, and tools for continuously monitoring the state of an application system and its usage, as well as for detecting, diagnosing, and resolving performance-related problems using monitored data.

This paper provides an overview of the state-of-the-art in common APM activities (Section 2) and tools (Section 3), and highlights related challenges and future directions (Section 4).

Previous works have been focused on one part of the work mentioned above. For example, previous papers can be used to describe the challenges of the APM discipline [5] or the challenges of the APM tools [6]. Our paper is the first to provide a comprehensive overview of the state-of-the-art in APM. The paper is organized as follows. Section 2 describes the basic concepts of APM and the classification of APM approaches. Section 3 details the evolution of APM specific data collection, storage, processing, and presentation. Section 4 discusses the challenges of APM and identifies research directions for future work.

2. APM ACTIVITIES
Regardless of the actual technical realization, APM is generally a set of activities that are periodically conducted and interconnected (activities) (Figure 1).

1. Data collection. Performance measures are collected from various sources. These sources can be either internal (e.g., a combination of complementary databases and log files) or external (e.g., third-party services).
2. Data storage and processing. The collected data is stored in a database and processed to make it available for visual inspection at different levels of abstraction and detail. This step can be triggered manually by a user, or it can be fully or automatically triggered based on rules set up on the system.

The remainder of this section details these activities.

2.1 Data Collection

Various types of performance-relevant measures can and must be collected to support the APM discipline. The classification of APM specific data collection is shown, what, when, and how often it is conducted, in the remainder of this section and detailed in Figure 2.

- 2.1.1. Where to Collect Data?
Modern application systems are multi-layered. Highly distributed environments are becoming more and more common. Therefore, APM specific data collection needs to be applied on clients and devices (e.g., third-party systems and human using desktop or mobile devices).

C. Heger, A. van Hoorn, D. Okanović, M. Mann:
Application performance management: State of the art and challenges for the future.
In: Proc. 8th ACM/SPEC ICPE, ACM (2017)

Application Performance Management (APM)

Kontinuierliche Überwachung von Anwendungsperformance



supported by

Inhalts-Erstellung:
Dr.-Ing. André van Hoorn (Prof.-Wrtz) und Stefan Siegl

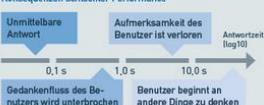
Einfluss von Performance auf Erfolg

Anwendungsperformance hat direkte Auswirkungen auf den Unternehmenserfolg.

> Z.B Wahrnehmung, Vertrauen, Umsatz



Konsequenzen schlechter Performance



Application Performance Management

Durch APM-Werkzeuge und -Prozesse kann Anwendungsperformance kontinuierlich überwacht und sichergestellt werden



1. Daten aus dem System abgreifen

Agenten sammeln Daten in sämtlichen Systemebenen. Auf der Anwendungsebene werden häufig technologie-spezifische Agenten eingesetzt.

	Aktiv	Passiv
Business	Umsatz, Conversion Rate, Bounce Rate	
Benutzer	Benutzerinteraktionen: Verweildauer, Ladezeiten, Fehler; Anzahl Ressourcen auf HTML-Seiten	
Anwendung	Komponentenfehleranfälligkeit, Methodenantwortzeiten, technische Fehlerzustände, Traces	
Middleware	Statistiken zu Warteschlangen, Pooling und Garbage Collection	
Betriebssystem	Statistiken zu File Handles, Threads, Virtualisierung	
Hardware	Auslastung von CPU, Memory, I/O etc.	

Für die unteren Systemebenen existiert eine Vielzahl standardisierter Schnittstellen.
> Z.B. JMX, Nagios

4. Interpretieren und Nutzen der Informationen

Aus den Informationen lassen sich manuell oder automatisiert Schlüsse ziehen und Aktionen ableiten.

Performance-Probleme erkennen. Typische Probleme sind erhöhte Anforderungen oder Ressourceneinsatz. Erkannt werden sie z.B. durch Vergleich mit Schwellenwerten oder „Normalverhalten“ (Baselines). Herausforderung: Falsch-Negativ/Positiv-Rate



Ursachen typischer Performance-Probleme erkennen, z.B. N+1-Problem, zu viele Remote-Calls, schlechte Datenbankabfragen. Diese (z.T. wiederkehrenden) Architektur- und Implementierungsfehler führt Anti-Patterns! Lassen sich insbesondere auf Basis aufgezeichnete Trace-Informations erkennen.

Automatisierte Aktionen: Insbesondere Cloud-Infrastrukturen bieten Dienste zur automatisierten Skalierung auf Basis der Monitoringinformationen an.

2. Von Daten zu Informationen

Es hat sich bewährt, Zeitreihendaten für das Monitoring zu erheben und diese mit detaillierten Traces für die Problemanalyse zu unterbauen.

Beispiele in APM-Werkzeugen

- > Solche Visualisierungs-Werkzeuge bieten in der Regel eine recht intuitive und schnelle Möglichkeit, Graphen, Metriken, Tabellen, etc. zu konfigurieren
- > Weitere Darstellungseigenschaften können dann konfiguriert werden [Legende, Axiem etc.]
- > Auf diese Weise können einzelne Panels angegeben werden



Trace	Antwortzeit
• df2f2e_1	(179 ms)
• searchTitleIdDescription_1	(173 ms)
• D432 ms	(D432 ms)
• searchTitleIdDescriptionWhQnWOrld_1	(252 ms)
• list_1	(142 ms)
• executeQuery@DB_Sales_FROD_0_1	(12 ms)
• executeQuery@DB_Sales_Me_0_1	(12 ms)

3. Visualisierung in navigierbaren Sichten

Aufgrund der hohen Menge müssen Informationen aufbereitet werden. Es hat sich bewährt, verschiedene Sichten auf die Daten anzubieten. Die Sichten sind navigierbar, um zu abstrakteren oder detaillierteren Sichten zur Beantwortung der jeweiligen Fragestellung zu gelangen.



APM-Werkzeuge

Übliche Funktionen von APM-Werkzeugen (Gartner-Dimensionen):

- > End user experience > Component deep dive
- > Architecture discovery > Resource Monitoring
- > Transaction profiling > Analytics
- > Kommerzielle Werkzeuge bieten umfassende Lösungen.
> Z.B. AppDynamics, BMC Software, CA APM, Dynatrace, HP Enterprise, IBM, New Relic, Riverbed Technology
- > Open-Source-Werkzeuge sind i. d. R. weniger mächtig, können jedoch Alternativen darstellen.
> Z.B ICINGA, insightP, Kieker, PinPoint, Zipkin, ZMon

APM-Wunschliste

Das Themenfeld APM hat sich in den letzten Jahren stark entwickelt. Es gibt aber noch genug Potenzial. Z.B.

Interoperabilität: Einige APM-Werkzeuge bieten einen Daten-Export. Eine erhöhte Interoperabilität (einheitliche Schnittstellen, Formate) ist wünschenswert.
> Z.B OpenTracing, OPENxtrace

Automatisierung: Manuelle APM-Aktivitäten (z.B. Konfiguration, Problemdiagnose) sind aufwändig und fehleranfällig.
Automatisierung, z.B. basierend auf Best Practices ist wünschenswert.

APM ist kein reines Technikthema. APM wird immer noch stark mit Technik assoziiert, ist aber längst auch ein Business-thema. Bis 2020 wird der Anteil des IT-Betriebs an den Käufern von APM-Lösungen lt. Gartner nur noch 40 % betragen (aktuell 75 %).

Technologie transfer: Forschung und Industrie können stark voneinander profitieren.
> Z.B Austausch von Traces, Probleme und Algorithmen, Kombination mit modell-basierten Ansätzen über Plattformen wie die „SPEC Research Group“.

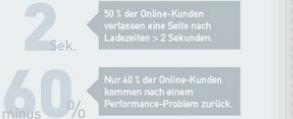
Application Performance Management

Kontinuierliche Überwachung von Anwendungen

Einfluss von Performance auf Erfolg

Anwendungsperformance hat direkte Auswirkungen auf den Unternehmenserfolg.

> Z.B. Wahrnehmung, Vertrauen, Umsatz



Konsequenzen schlechter Performance



Application Performance Management

Durch APM-Werkzeuge und -Prozesse kann Anwendungsperformance kontinuierlich überwacht und sichergestellt werden

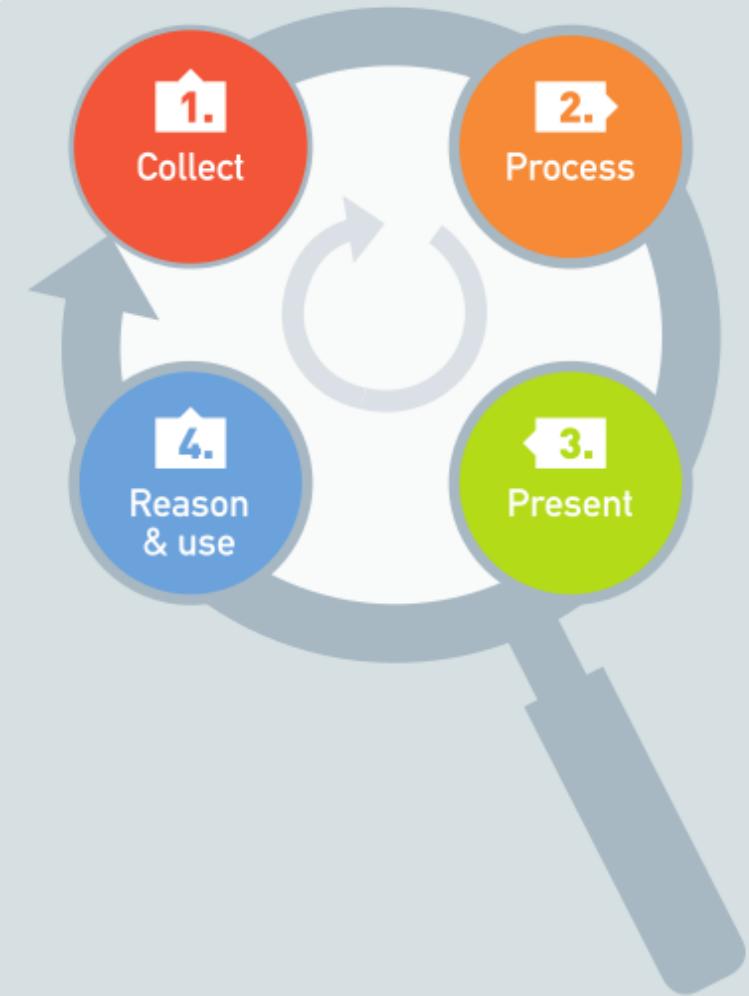
... wichtig im gesamten Software-Lebenszyklus



... relevant für alle Systemebenen



... erfordert Zusammenarbeit aller Abteilungen



by

Inhaltsliche Entwicklung:
Dr.-Ing. André von Hehn (Prof.-Wtrr.) und Stefan Siegl



mit detaillierten Traces für die

und schnelle Möglichkeit, Graphen, Metriken, Tabellen, etc. zu konfigurieren gegeben, die spezifizieren, welche Zeitreihen visualisiert werden sollen enden, Axen etc.]

Trace	Antwortzeit
v= df2User_1	(1792 ms)
+ searchTitleIdDescription_1	(1792 ms)
+ searchTitleIdDescriptionWZQueryWordL_1	(1792 ms)
v= list_1	(1792 ms)
+ searchQuery@DB:Select FROM _0_	(1792 ms)
+ searchQuery@DB:Select Me_0_	(1792 ms)
+ ...	-

ten

tet werden. Es hat sich bewährt, verschiedene ar, um zu abstrakteren oder detaillierteren erlängen.

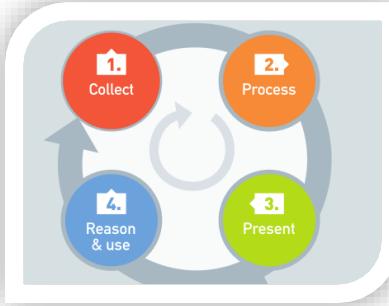


ma.
mit Technik
heit des IT-Ber-
M-Lösungen
ragen (aktuell

Technologietransfer:
Forschung und Industrie können stark voneinander profitieren.
> Z.B. Austausch von Traces, Probleme und Algorithmen, Kombination mit modell- basierten Ansätzen über Plattformen wie die „SPEC Research Group“.

1.

Collecting Data from All System Levels



- Agents collect data from all system levels
- On application level the agents are often technology-dependent

Where?

What?

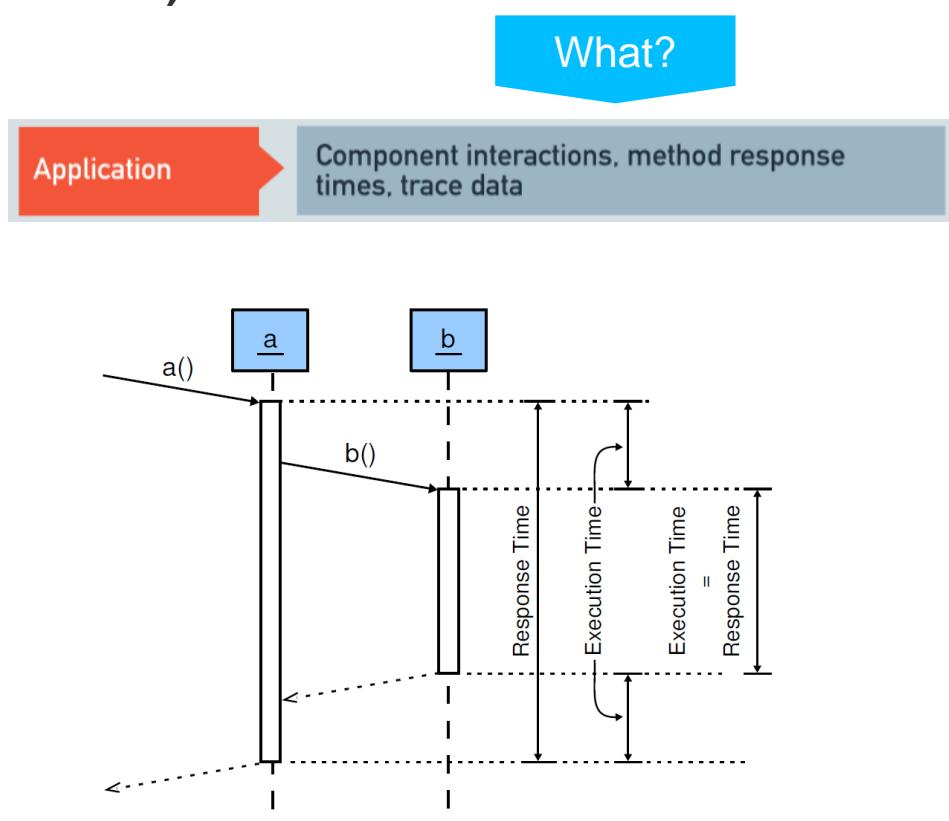
How?

Business	Sales data, conversion and bounce rate	Active	Passive
User	User interactions: length of stay, load times, errors; number of resources on HTML pages	Stimulation of the system by periodic requests. E.g., synthetic user transactions	Collection of runtime data from real system usage. E.g., injection of code, analysis of network traffic, resource utilization, or log files
Application	Component interactions, method response times, trace data		
Middleware	Queuing statistics, pooling, garbage collection		
Operating System	File handling statistics, virtualization, thread statistics		
Hardware	CPU load, memory consumption, I/O statistics	Some technologies on lower levels provide standard interfaces for data collection, e.g.. Nagios, JMX	

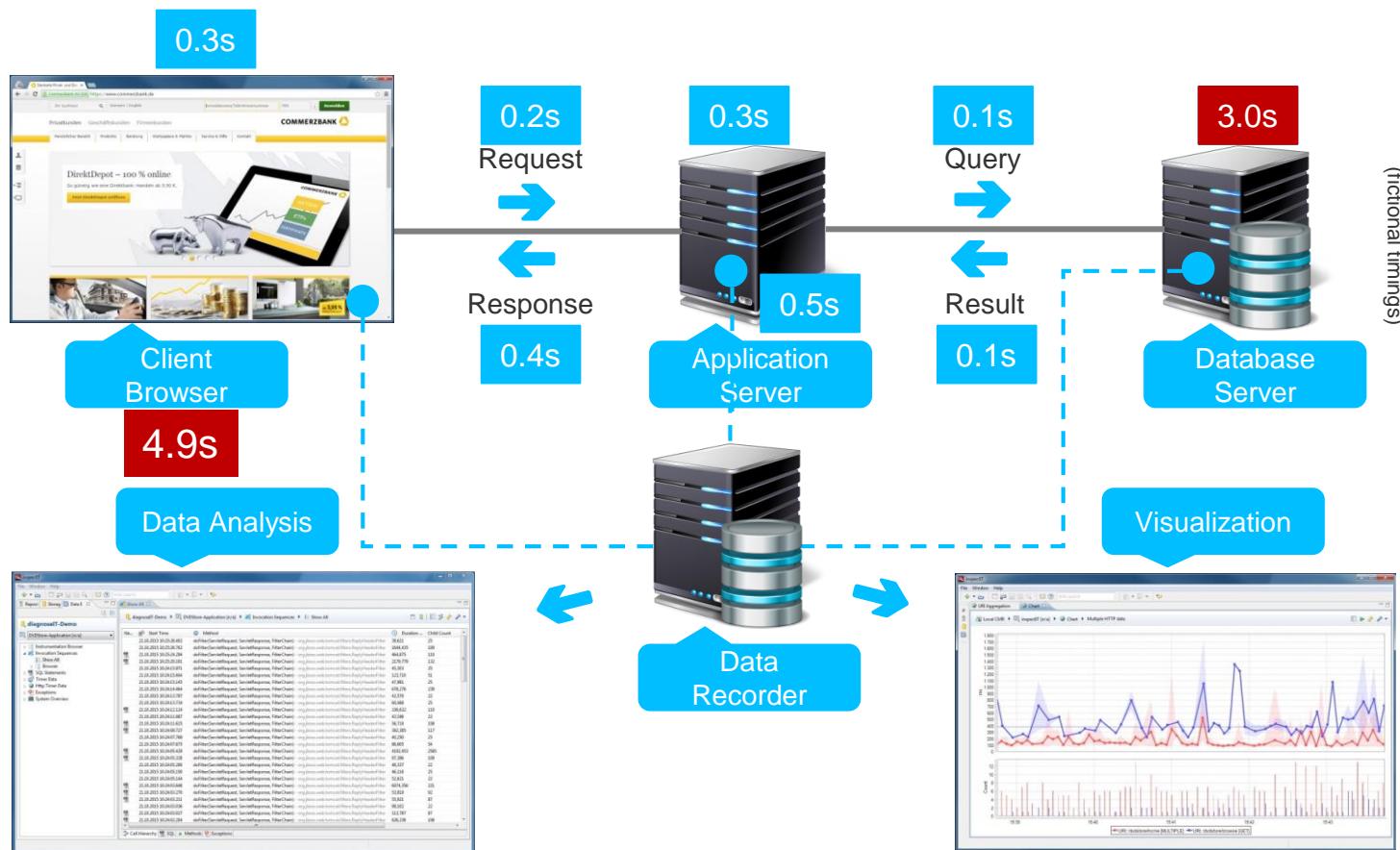
Trace-based Metrics (Selection)

What?

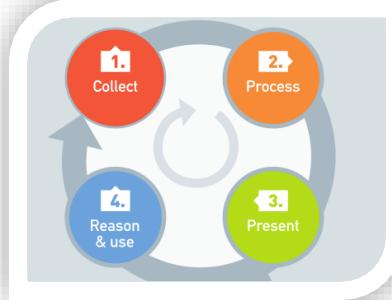
Metric
Response Time
CPU Time
Method Name
Return Type
Logging Level
SQL Statement
Error Message
...



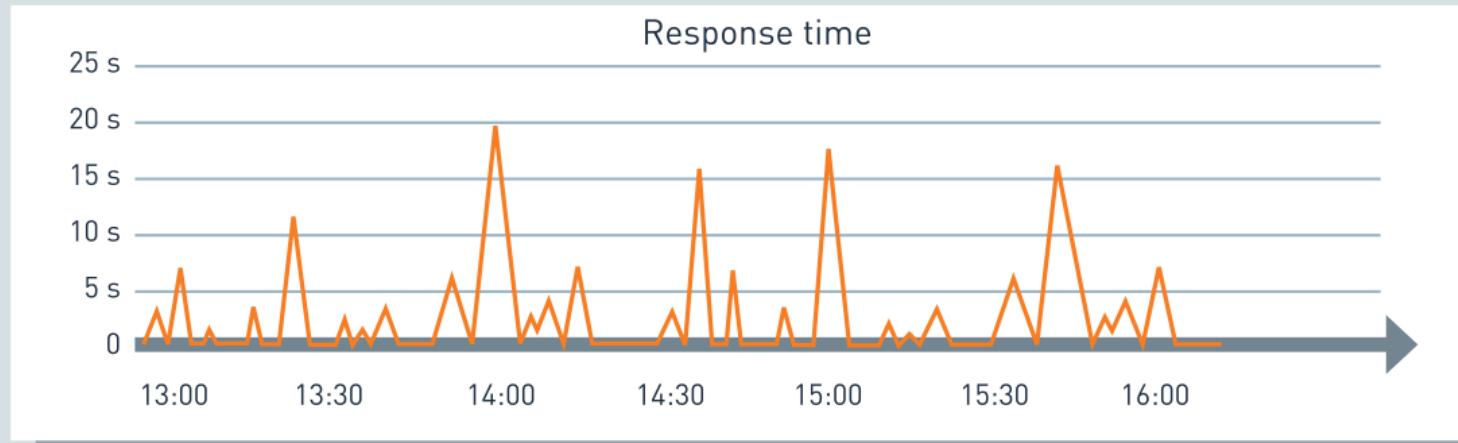
Monitoring (Measurement-based Performance Evaluation)



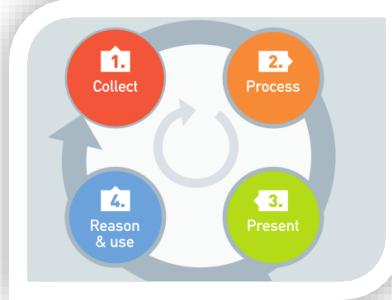
Reconstructing Information from Data



- Data is collected from the system...
- represented as **time series**...



Reconstructing Information from Data



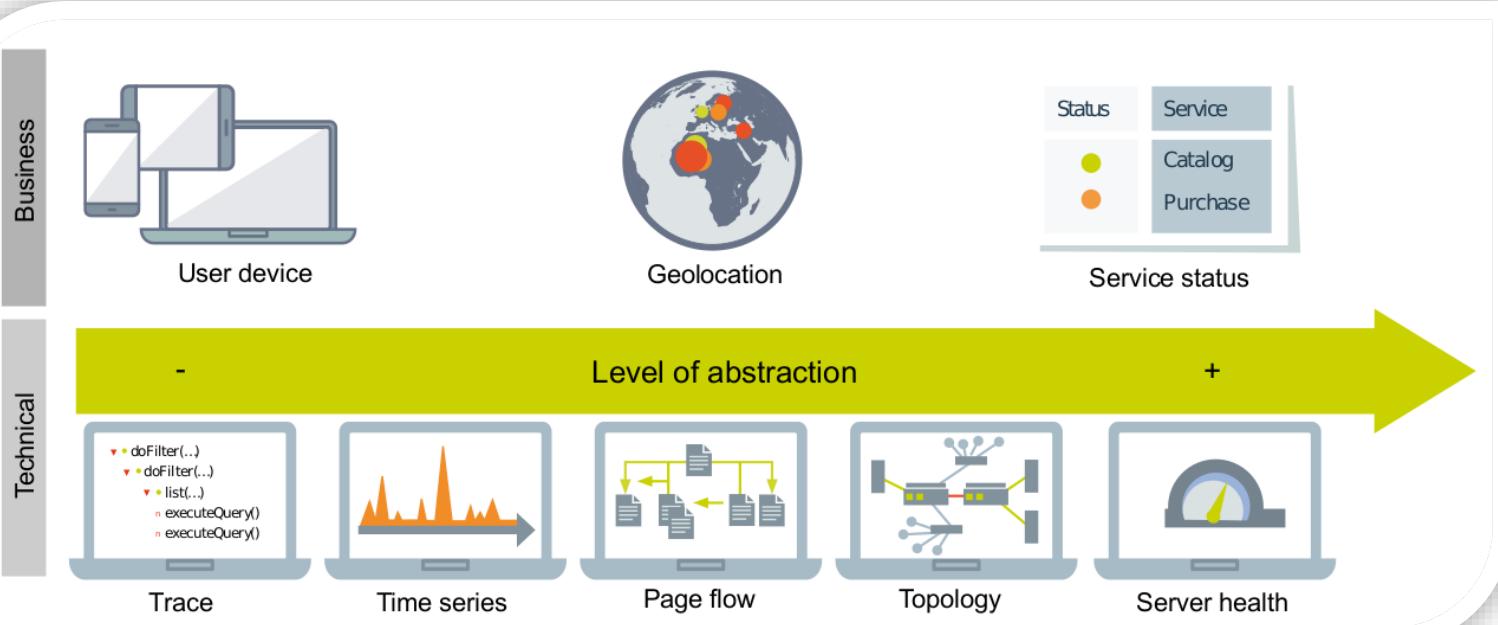
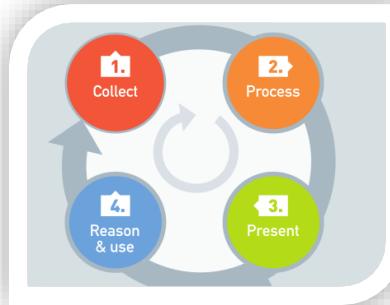
- Data is collected from the system...
- represented as **time series**...
- ... and as **detailed execution traces**, and used to support problem analysis

Trace	Response time
▼ ● doFilter(...)	[1793 ms]
▼ ● searchTitleAndDescription(...)	[1730 ms]
▼ ● searchTitleAndDescriptionWithOneWord(...)	[1632 ms]
▼ ● list(...)	[352 ms]
■ executeQuery() {SQL: Select PROD_ID ...}	[143 ms]
■ executeQuery() {SQL: Select INV_ID ...}	[12 ms]
▼ ●
...	...
...	...

3.

Visualization Through Navigable Views

- High quantity of information has to be pre-processed
- It has proven useful to use **different views** to show the data
- **Views are navigable** and can be categorized by both scope and detail level

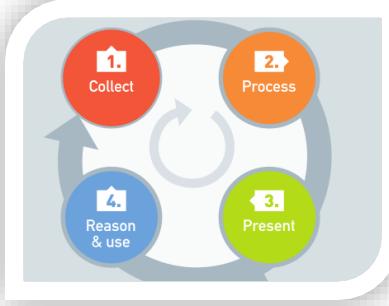


Example: Application Topology Discovery and Visualization



© AppDynamics

Interpreting and Using the Information



Manual or automated conclusions and actions can be derived from the information, e.g.,

- **Problem detection and alerting**

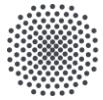
- E.g., increased response times and resource utilization
- Detection, for instance, based on thresholds and baselines

- **Problem diagnosis and root cause isolation**

- E.g., N+1 problem, too many remote calls, poor DB queries
- Detection based on monitoring information

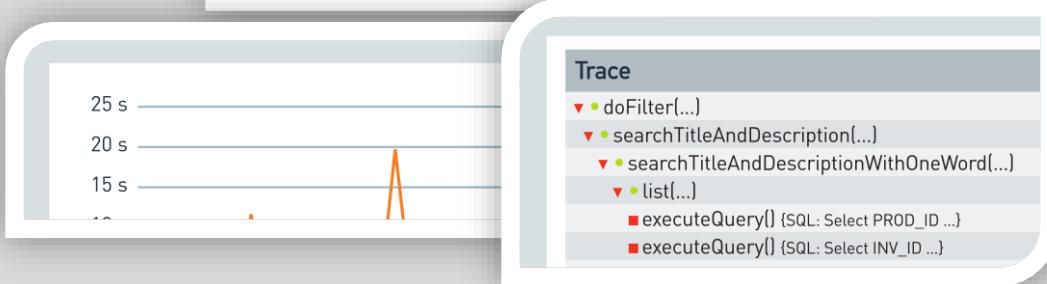
- **System refactoring and adaptation**

- E.g., auto-scaling in cloud-based architectures



State of the Art of Visualization in APM Tools

Part 2/2: Examples Visualizations in APM Tools



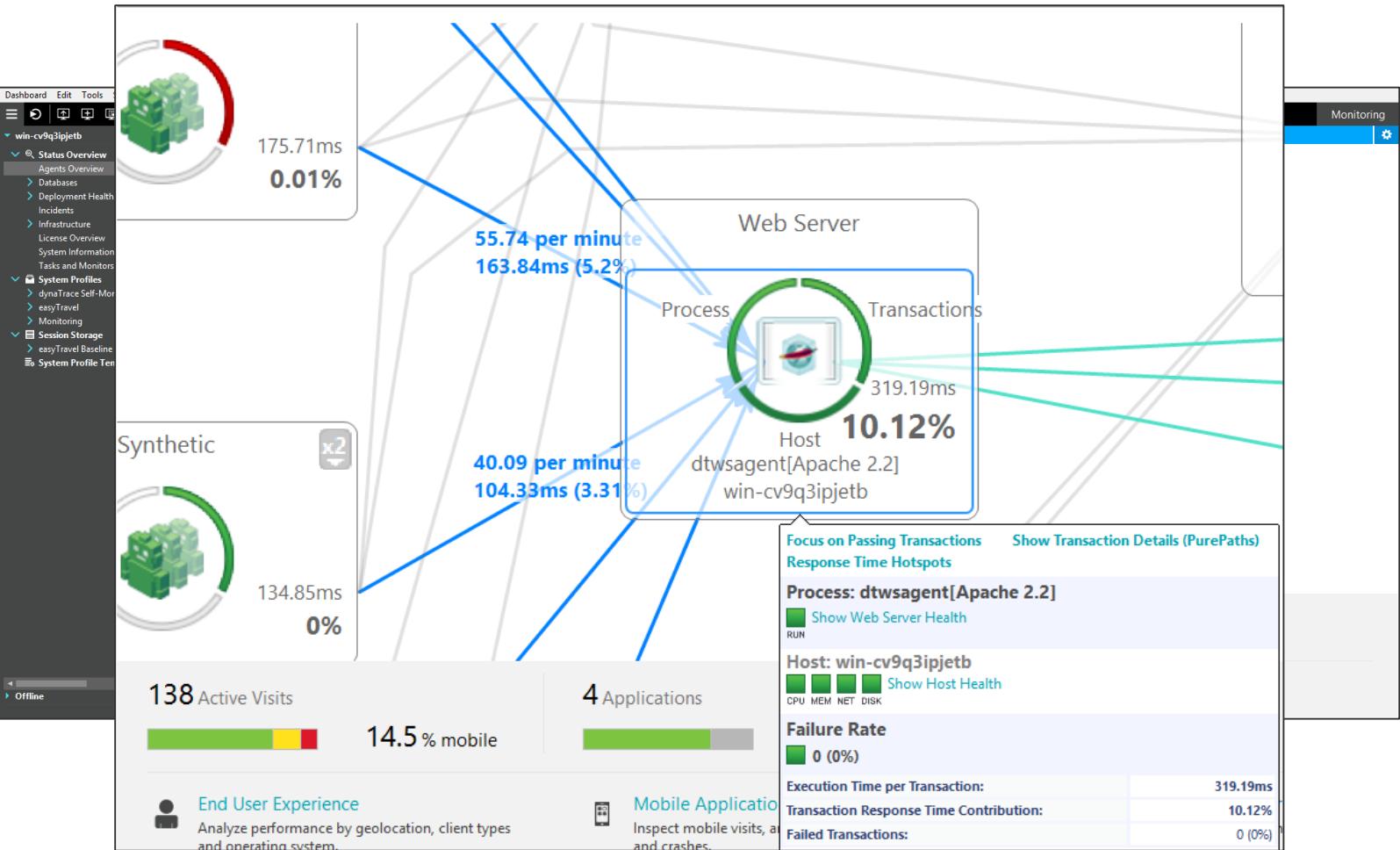
André van Hoorn
Dušan Okanović

Commercial APM Tools

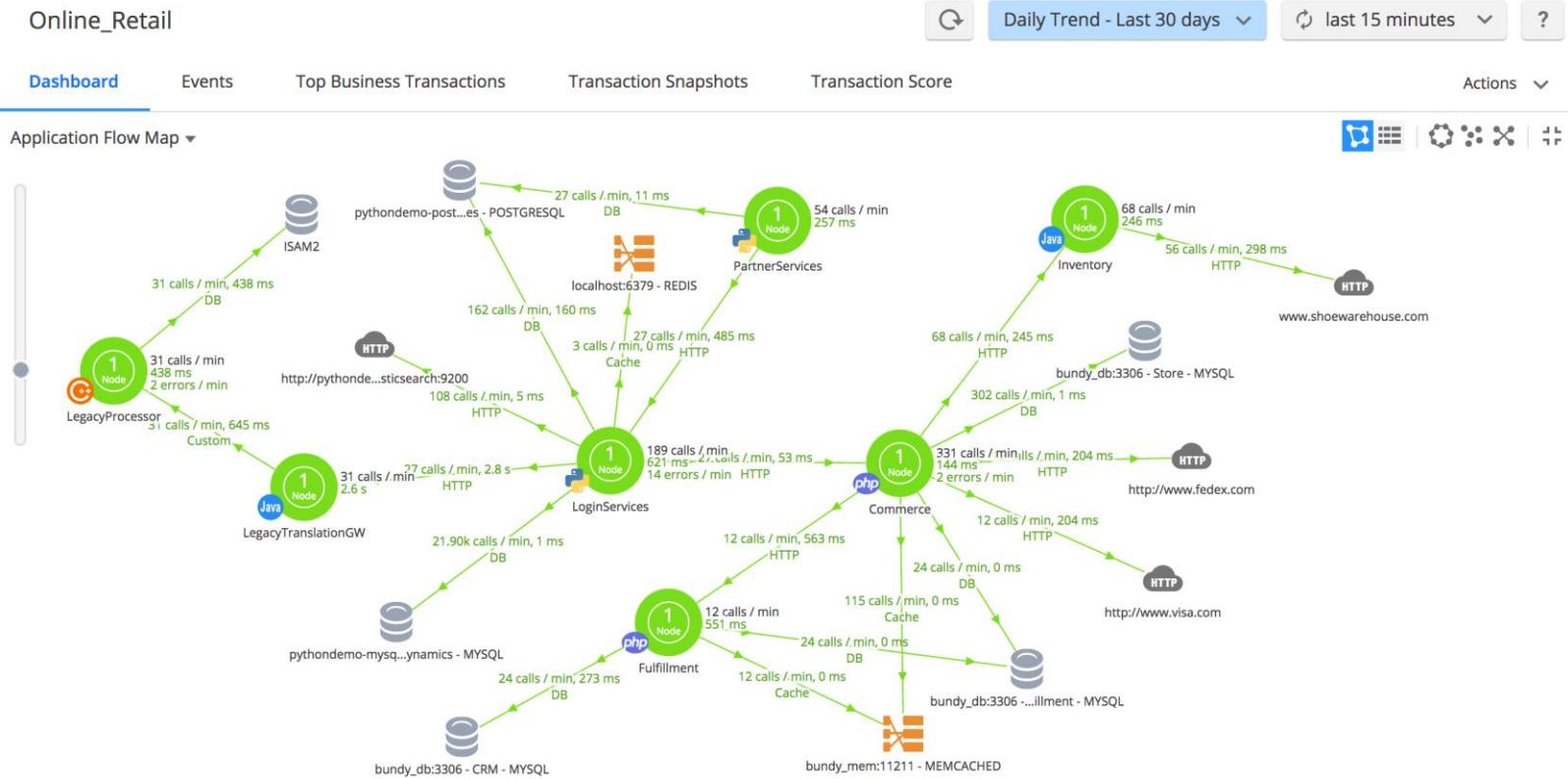
Magic Quadrant



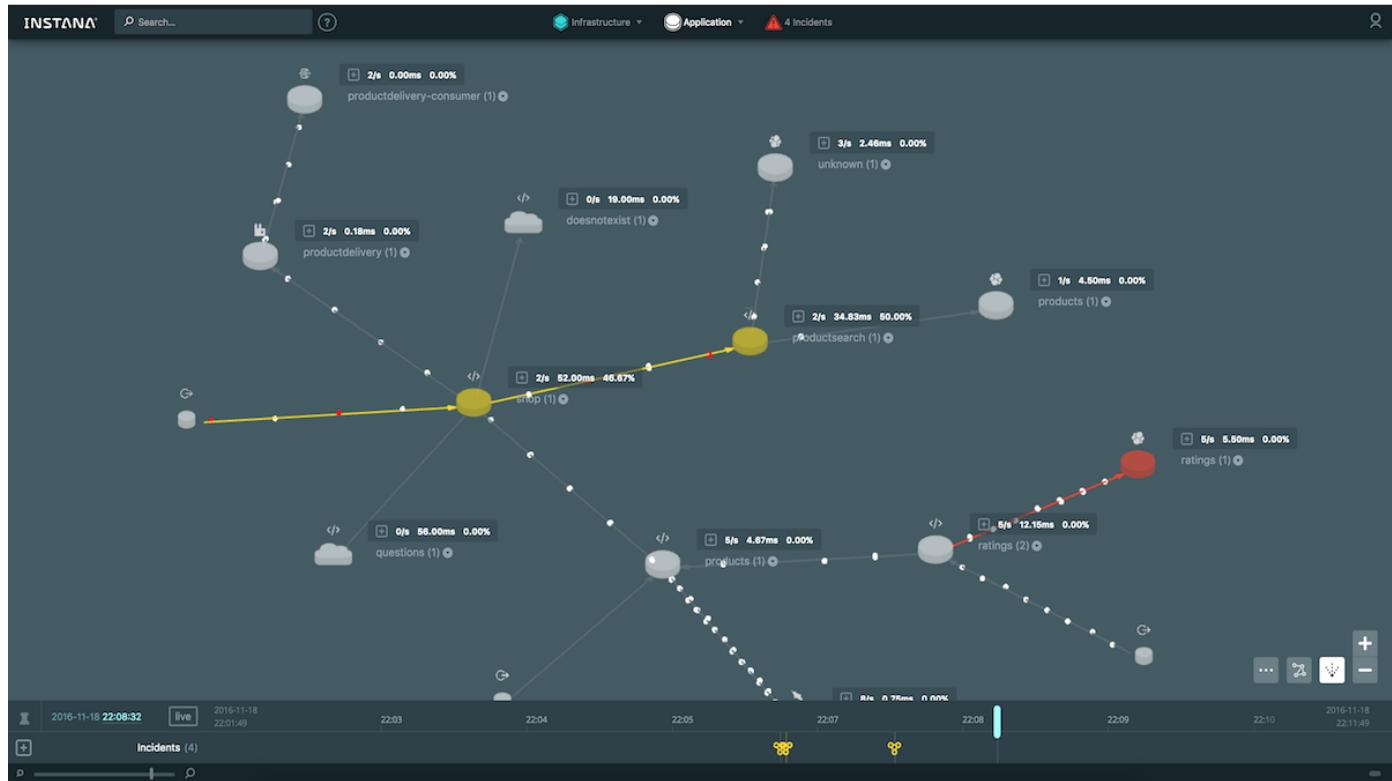
Application Overview – Dynatrace



Application Overview – AppDynamics



Application Overview – Instana



End User Monitoring – AppDynamics

Ecommerce-iOS



last 15 minutes

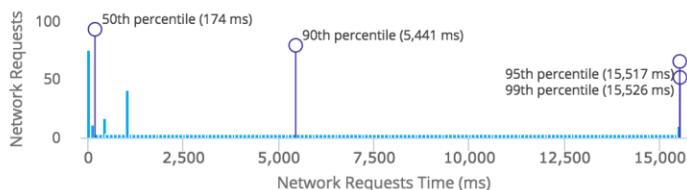


Overview

Geo Dashboard

Usage Stats

Network Requests Time Distribution



Total Crashes

2

Total Crashes

Crash Rates by App Version

2.0	5.9%
2.1	5.9%
1.0	-

App Opens by Country >



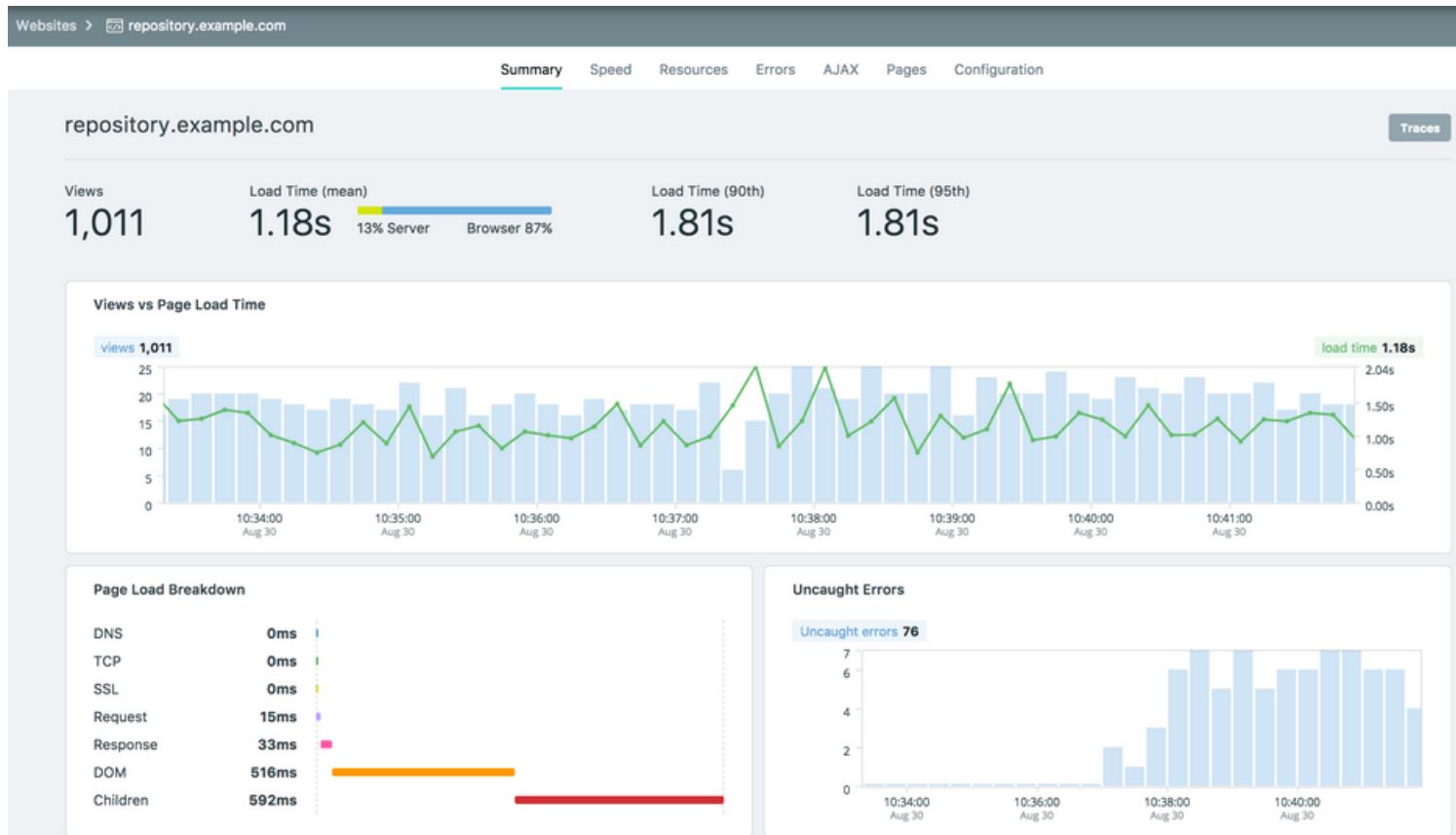
App Opens Per Minute



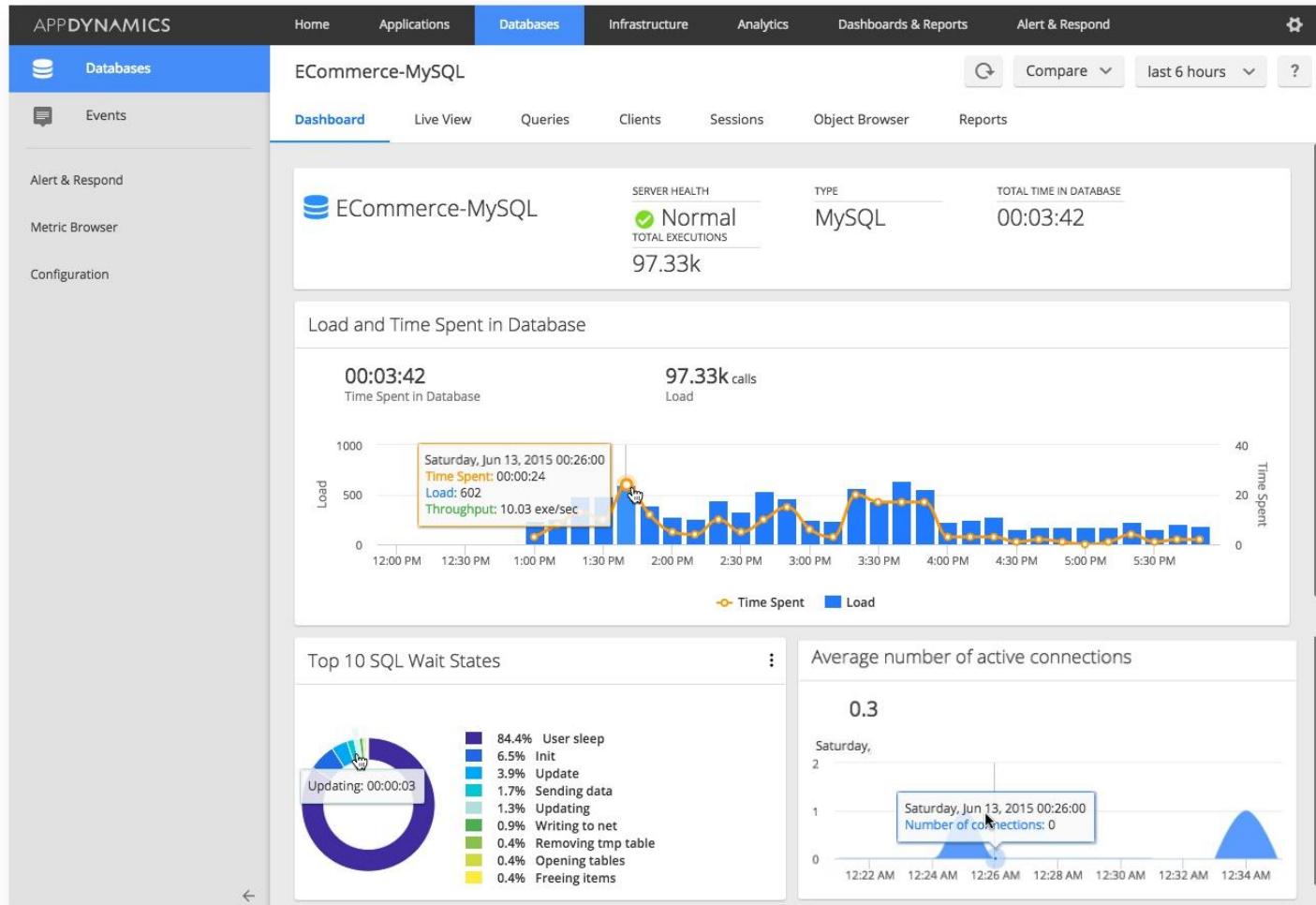
Top 5 Unique Crashes

SIGSEGV in promovie	1 Crashes	1 Impacted Users
SIGSEGV in add	1 Crashes	1 Impacted Users

End User Monitoring– Instana



Database Monitoring – AppDynamics



Server Monitoring – AppDynamics

APPDYNAMICS

Home Applications User Experience Databases Servers Analytics Dashboards & Reports Alert & Respond

Servers

TIER1TOMCAT

Service Availability Events Alert & Respond Metric Browser Configuration

Dashboard Volumes Network Processes

Availability

100.0 %

CPU

1 CPUs 4 Cores 13.0 % Usage

Memory

15.6 GB Installed 98.1 % Usage 13.0 % Swap Usage

Network

2 Interfaces 38.0 kB/s Incoming 4.2 kB/s Outgoing

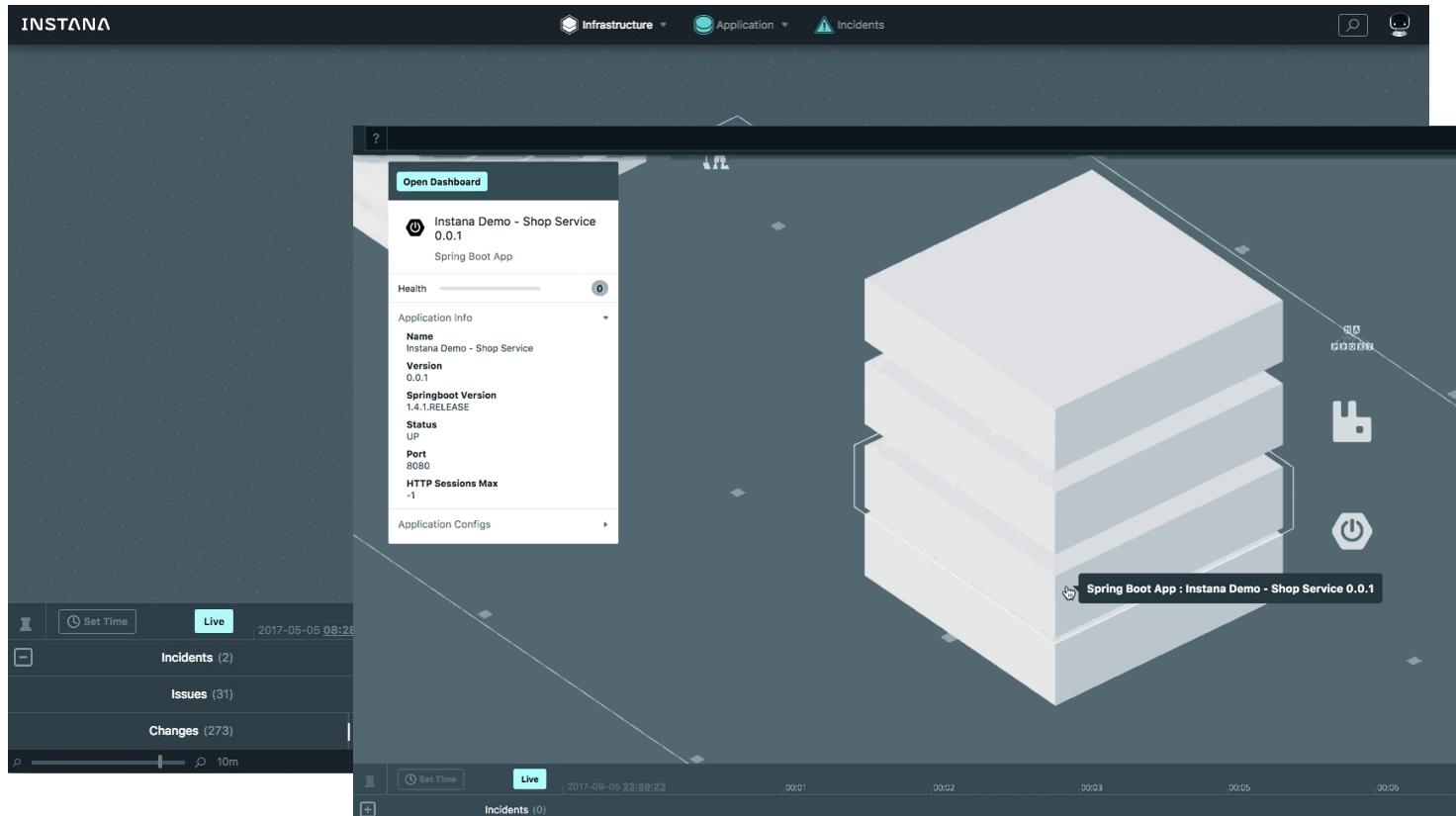
Properties

AppDynamics|Agent|Machine Info os.name=Linux|os.arch=amd64|os.version=2.6.32-431.el6.x86_64
AppDynamics|Agent|Install Directory /build/cart/TIER1TOMCAT/machineagent
AppDynamics|Agent|Build Number 315339d

Volumes

Name	Partition	Total(MB)	Free(MB)	User... (%)	Usage (%) Trend
/boot	sda1	485	421	8	
/	dm-0	199,494	105,776.5	42	

Server Monitoring – Instana



Problem Identification – Dynatrace

easyTravel > All applications > Applications > PurePaths

Last 24 hours

/Booking

Open in client

All

Response time

- Very slow
- Slow
- Normal
- Fast
- Very fast

State

- No errors
- Detected errors
- Failed

Database

- DB chatty
- Single long SQLs
- N+1 query problem
- Overall DB time too high

Web service

- High payload
- Chatty
- Slow
- N+1 calls
- High web service time

Threading

Response time

Threading

Complexity

Database

Top findings

Response time	738ms (Normal)	Database	N+1 query problem
Complexity	Medium	Web service	Slow
Threading	Heavy	HTTP	-
Async	Normal	Error cause	-

Breakdown

739MS Duration

738MS Response time

1s 95ms Exec time

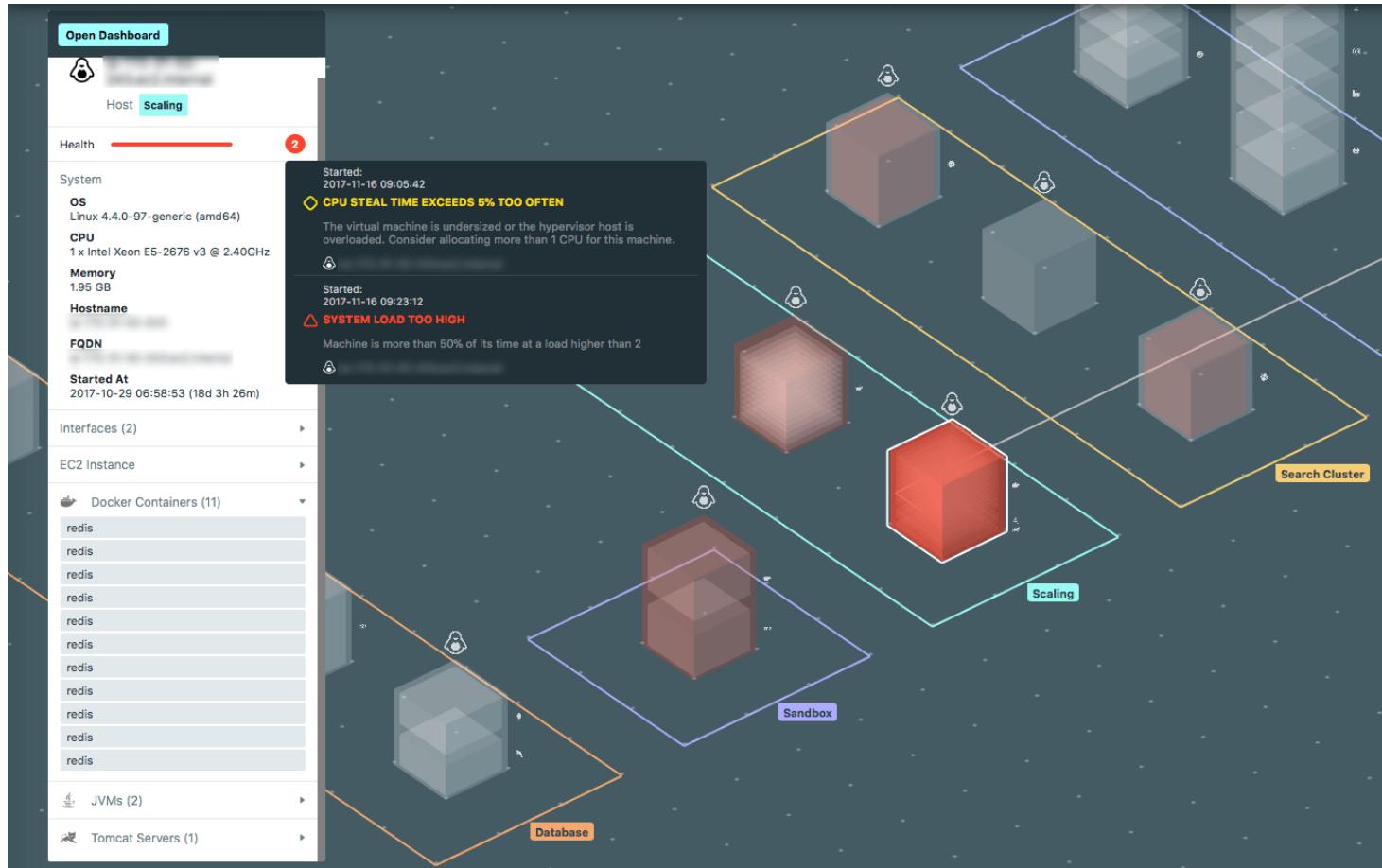
155ms CPU, 337ms Sync, 274ms IO, 327ms Suspension

You are viewing the 100 most recent PurePaths

Filter PurePaths

State	PurePath	Start time	Exec time	Response time	Nodes
	/Booking	2016-12-05 16:34	1s 95ms	738ms	163
	/contact-orange.jsf	2016-12-05 16:34	37ms	37ms	16
	/legal-orange.jsf	2016-12-05 16:34	27ms	27ms	11
	/contact-orange.jsf	2016-12-05 16:34	23ms	23ms	3
	/contact-orange.jsf	2016-12-05 16:34	12ms	12ms	3
	/Scripts/jquery.easing.1.3.js	2016-12-05 16:34	10ms	10ms	4

Problem Identification - Instana



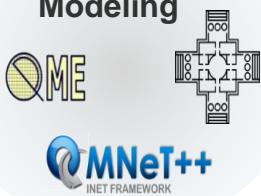
Open Source APM tools

Open Source APM tools

Monitoring & Application Deep Dive



Performance Modeling



Real User Monitoring



Low-Level Performance Profiling



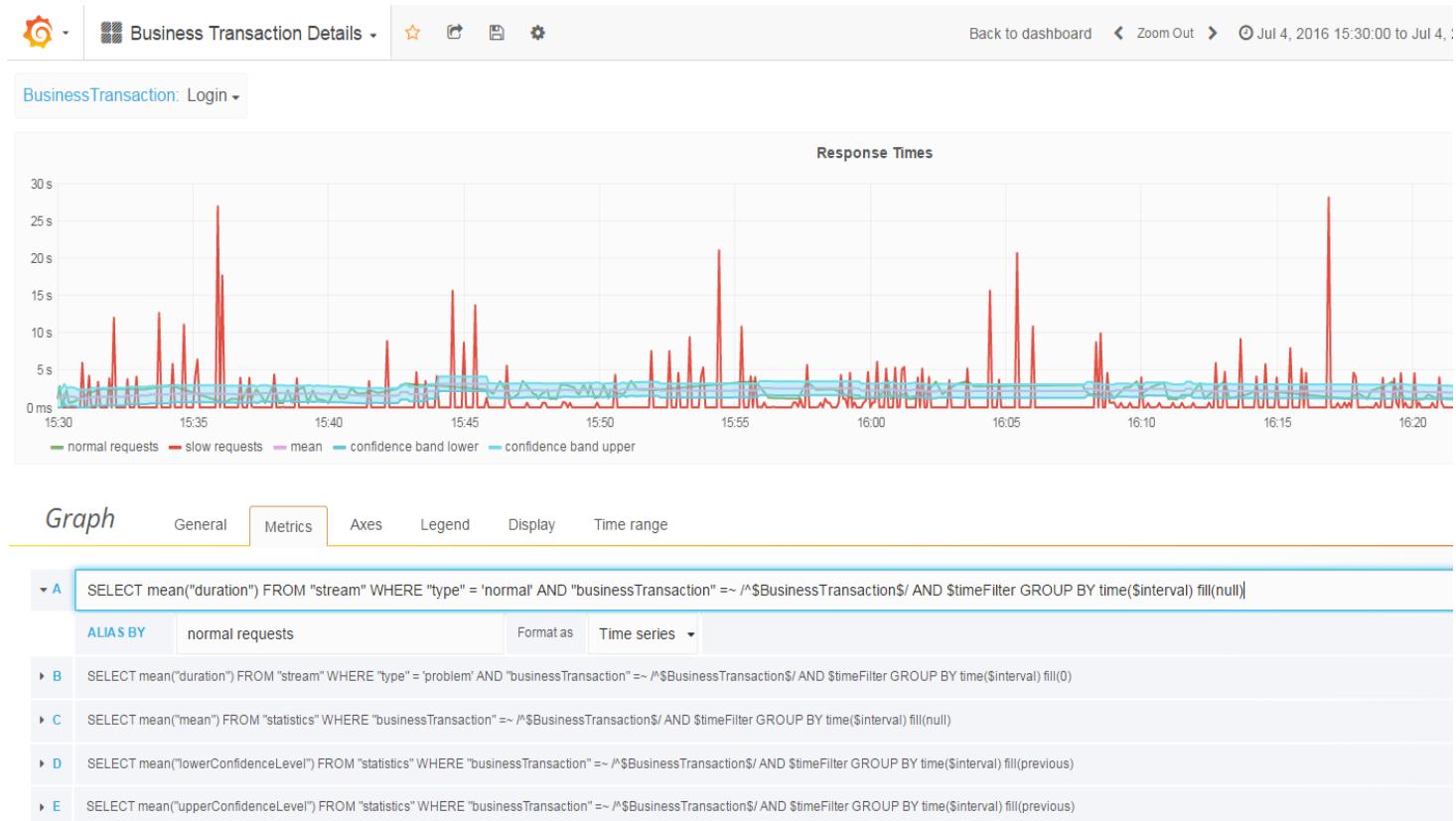
System & Resources Monitoring



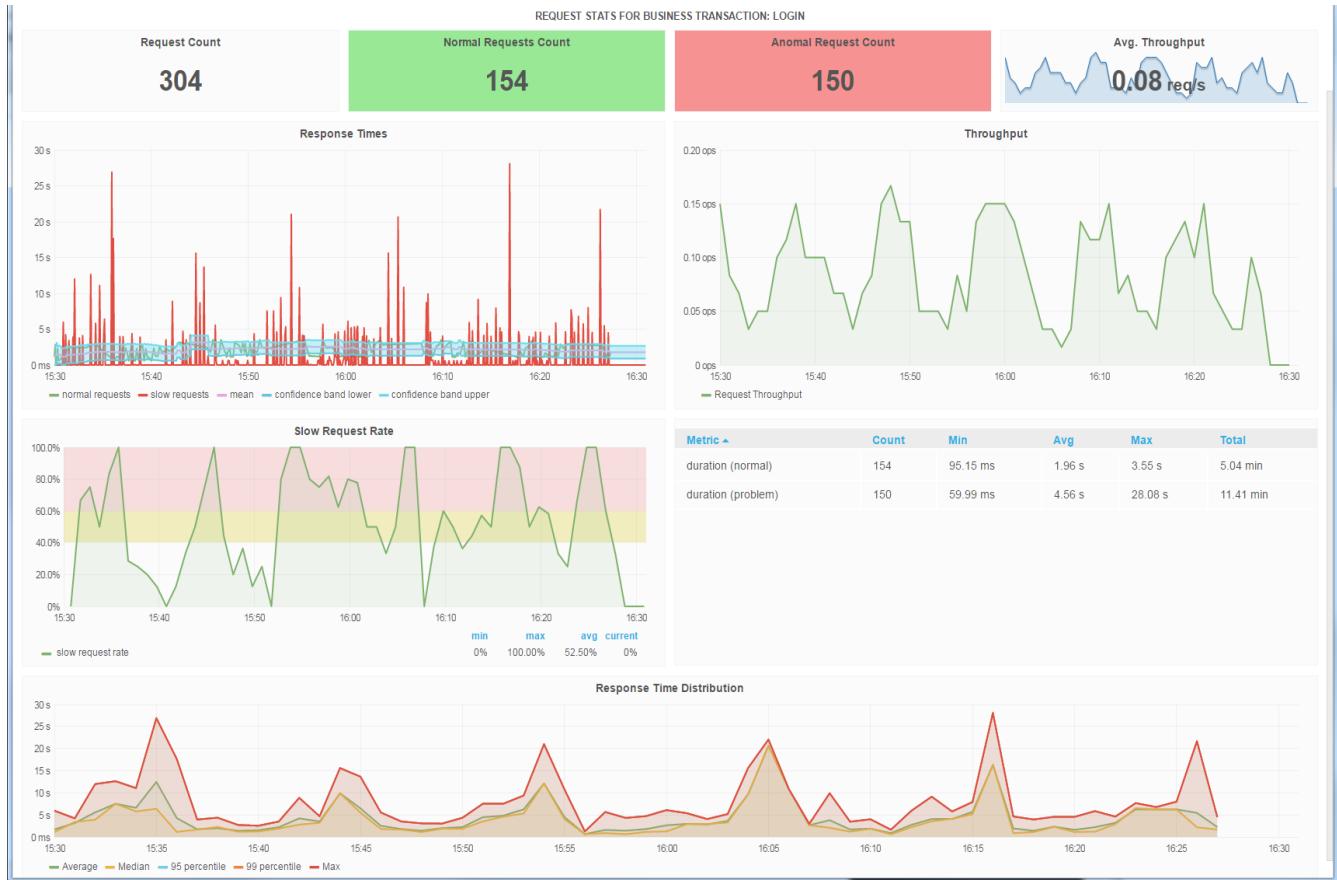
Web Performance Analysis



Dashboards – Grafana



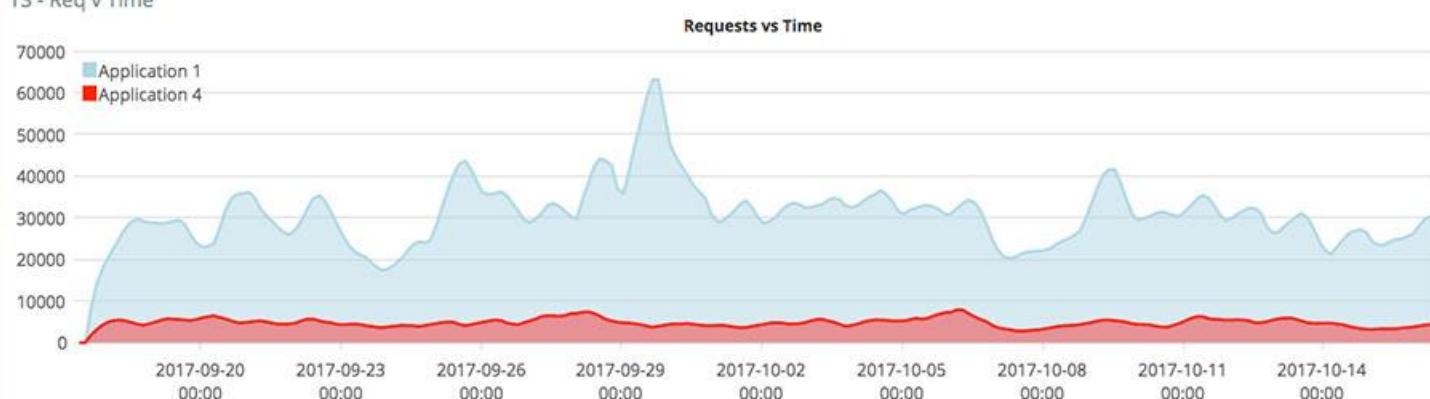
Dashboards – Grafana



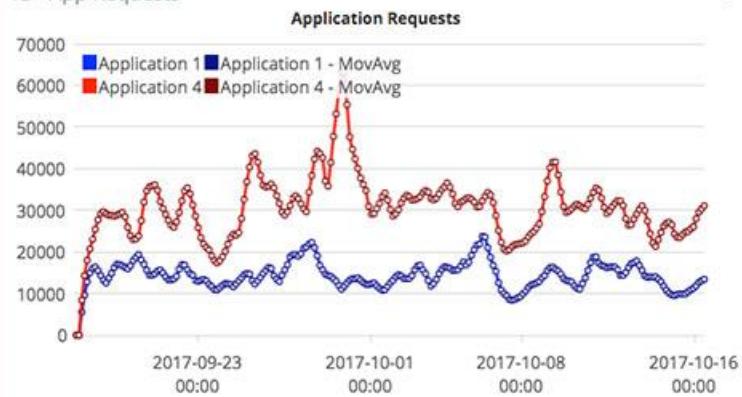
Dashboards – Kibana



TS - Req v Time



TS - App Requests



TS - App4

TS - App1



Dashboards – Kibana



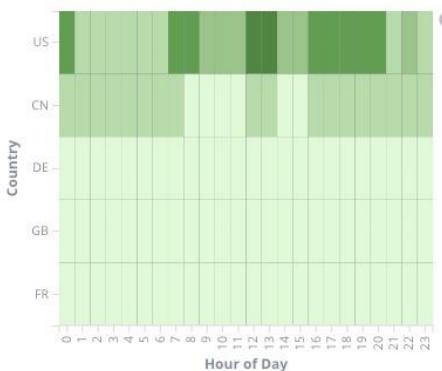
Apache - Total Visitors

4,931,584

Apache - Unique Visitors

29,740

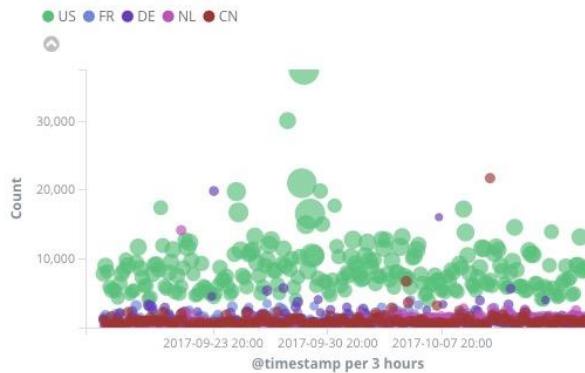
Apache - Country traffic by hour



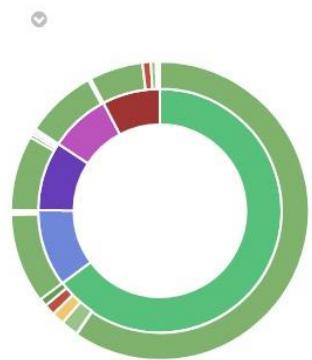
Apache - Unique Visitors ...

City	Unique Visitors
Beijing	562
Redmond	445
Ashburn	400
Chicago	373
Los Angeles	245
Seattle	233
San Jose	232
Singapore	208

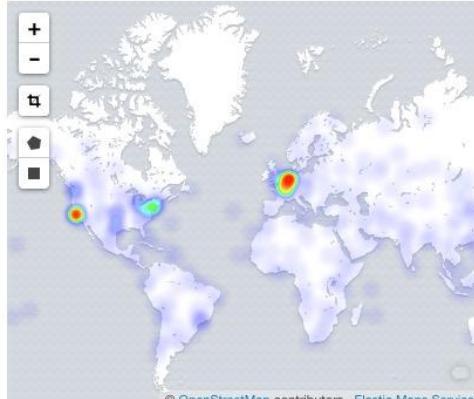
Apache - Bytes and Count



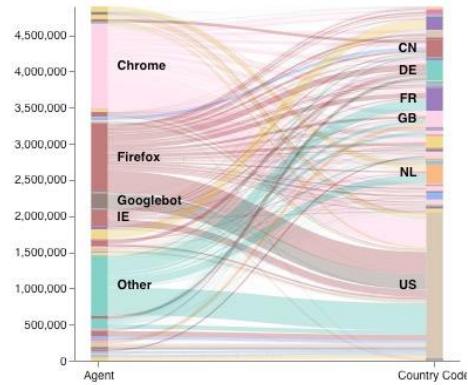
Apache - Country and Status



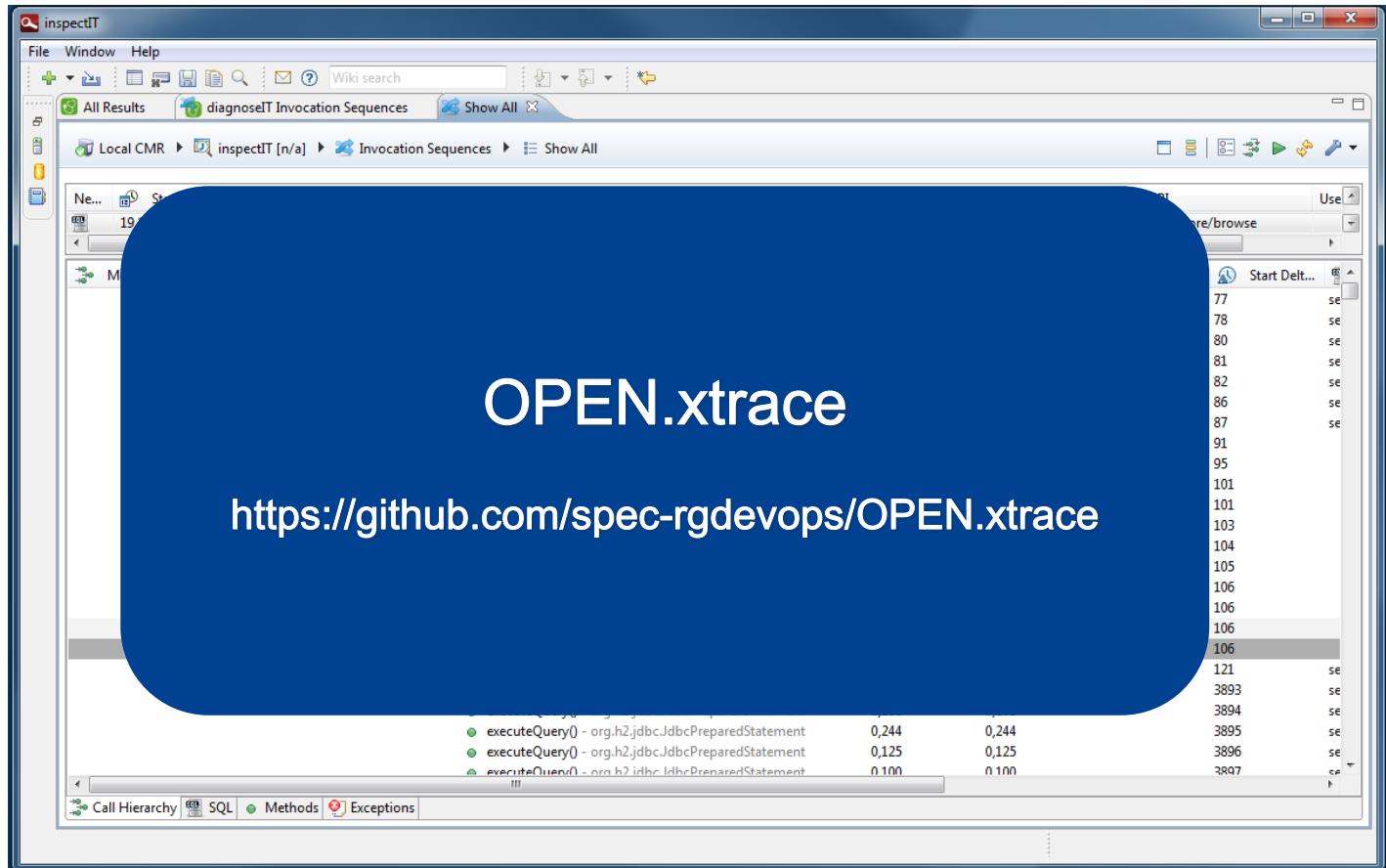
Apache - Visitor Map (geocentroid)



Apache - Browser to Country (vega)



Execution Traces – inspectIT



Tracing – Zipkin

Zipkin Investigate system behavior

Find a trace

Aggregates

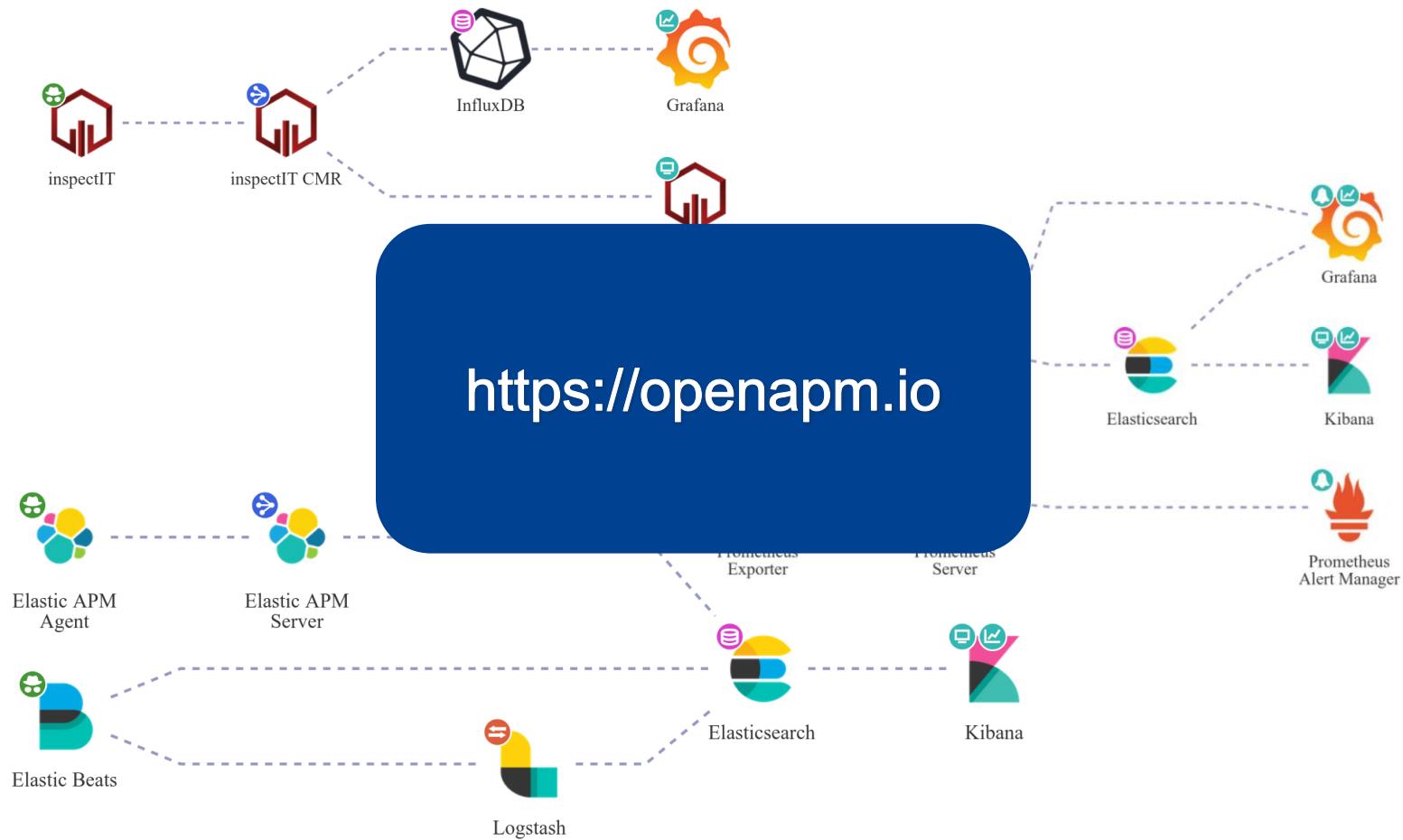
Services

317.000ms

634.000ms

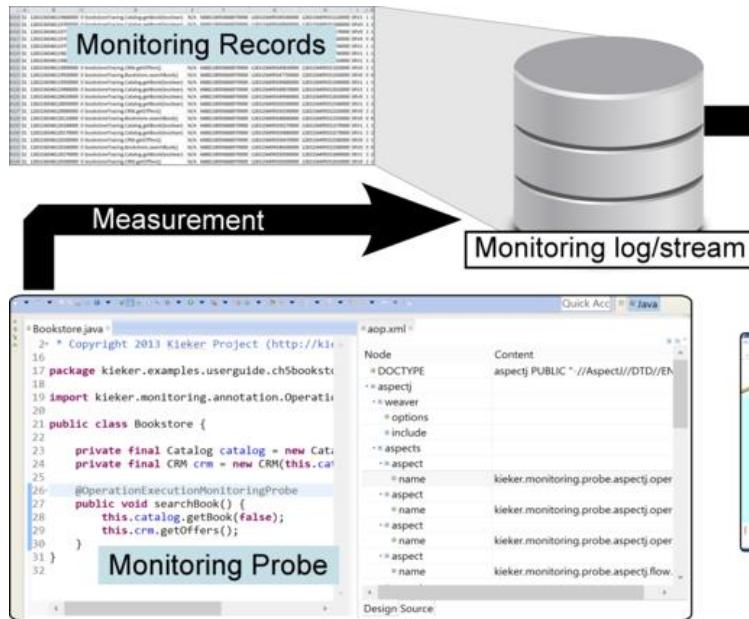
- c*:Test Cluster:local	1.585s : QUERY	
c*:Test Cluster:local	1.000ms : Parsing select * from vortex_powervomit_test.envelopes limit 100; [SharedPool-Worker-1]	
c*:Test Cluster:local	1.000ms : Preparing statement [SharedPool-Worker-1]	
c*:Test Cluster:local	2.000ms : Computing ranges to query [SharedPool-Worker-1]	
c*:Test Cluster:local	15.000ms : Submitting range requests on 257 ranges with a concurrency of 1 (24892.65 rows per range expected)	
c*:Test Cluster:local	69.000ms : Executing seq scan across 7 sstables for (min(-9223372036854775808), min(-9223372036854775808))	
c*:Test Cluster:local	531.000ms : Submitted 1 concurrent range requests covering 257 ranges [SharedPool-Worker-1]	
c*:Test Cluster:local	2.000ms : Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	1.000ms : Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	: Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	: Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	: Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	1.000ms : Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	: Read 12 live and 0 tombstone cells [SharedPool-Worker-2]	
c*:Test Cluster:local	: Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	: Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	: Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	: Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	8.000ms : Seeking to partition beginning in data file [SharedPool-Worker-2]	
c*:Test Cluster:local	1.000ms : Read 12 live and 0 tombstone cells [SharedPool-Worker-2]	

Build Your Own Landscape



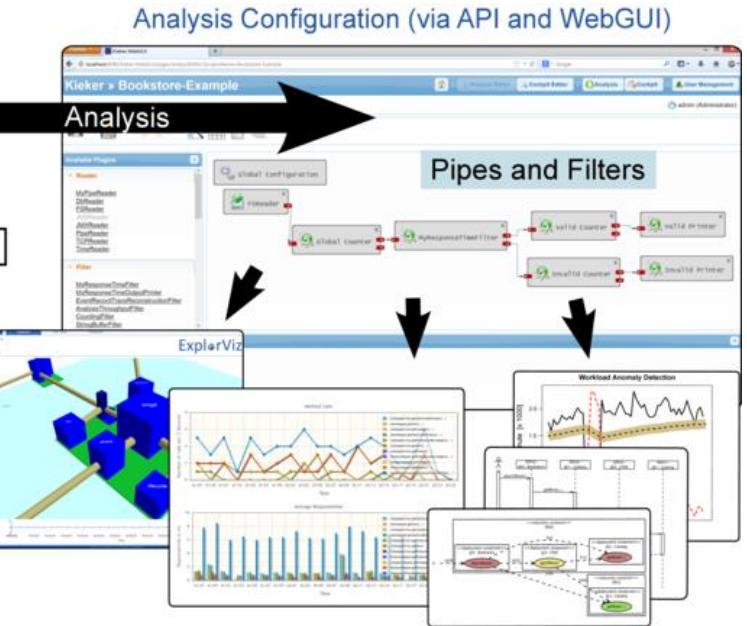


Dynamic Software Analysis and Application Performance Management

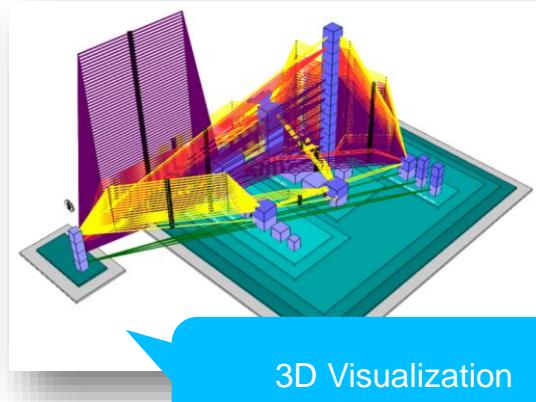
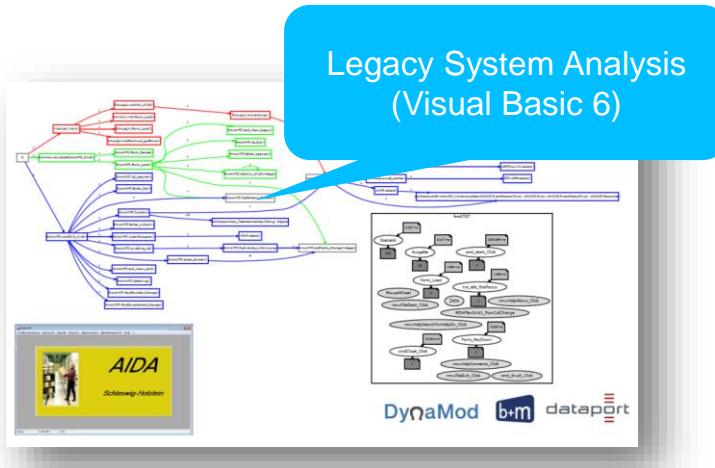
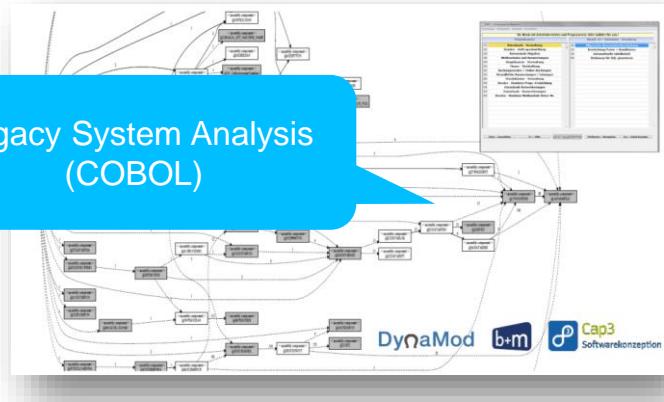
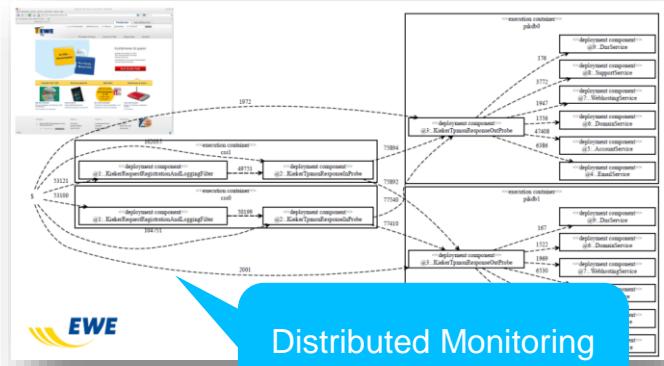


Software System with Monitoring Instrumentation

- <http://kieker-monitoring.net>



Application (and Visualization) Examples



Conclusion

- Commercial tools have most fancy visualizations
 - ... how useful are they really?
- Open-source tools are flexible and can be adapted as needed
 - ... but require a lot of work to setup
- Thanks to a more widespread adoption of (open-source) APM, it is easier to get access to rich APM data (e.g., distributed traces) and to integrate visualization approaches (e.g., via APIs)
 - ... if you know what and how to visualize