# The Differing Ways to Monitor and Instrument

***Southern Computer Measurement Group***

Jonah Kowall, VP Market Development and Insights

Twitter : @jkowall

APPDYNAMICS

# Jonah Kowall's Background

- 23 years in IT
- Infrastructure and Operations enterprises and startups (17 yrs)
  - Security - CISSP, CISA, PCI
  - Started one of the first content filtering companies
- Head of global monitoring at Thomson Reuters
- Head of IT Operations at MFG.com – Bezos Expeditions
- Gartner Research VP 3.5 years
- Strategy AppDynamics 1.5 years

# Agenda

- Introduction to Instrumentation
- Instrumentation done by people
- Challenges in instrumentation of the right processes
- Technical instrumentation of Browsers and Mobile
- Technical instrumentation of Server (php, Java, .NET…)
- How to implement the right logging
- How to correlate across technologies

# Definitions

*Instrumentation*

“The design, construction, and provision of instruments for measurement, control, etc; the state of being equipped with or controlled by such instruments collectively.”

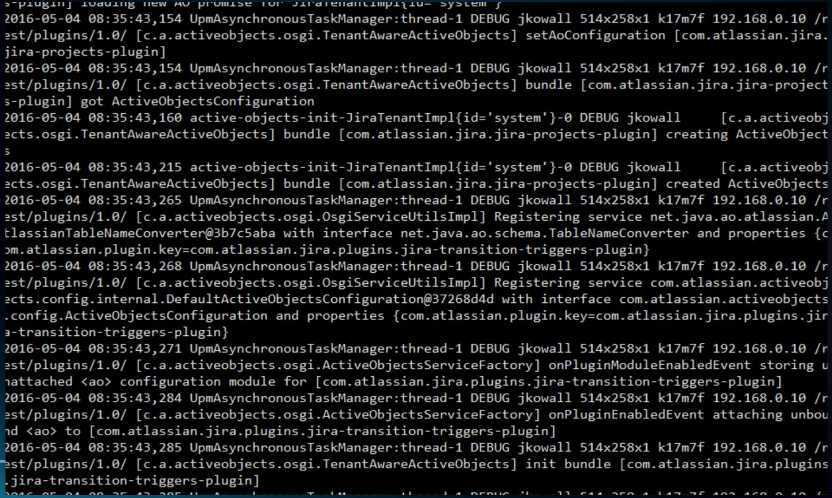
*Telemetry*

“Automated communications process by which measurements are made and other data collected at remote or inaccessible points and transmitted to receiving equipment for monitoring.”

# Software Instrumentation Data Types

- Metrics
  - Key value pairs
  - Numeric values
  - Time series

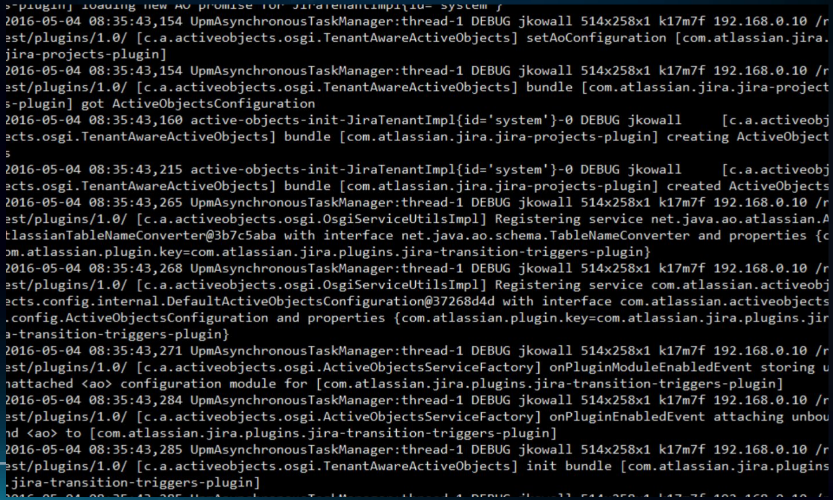
- Events
  - Informational
  - Errors
  - Critical Events



```
s-plugin] loading new AO promise for JiraTenantImpl{id='system'}
2016-05-04 08:35:43,154 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.TenantAwareActiveObjects] setAoConfiguration [com.atlassian.jira.
jira-projects-plugin]
2016-05-04 08:35:43,154 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.TenantAwareActiveObjects] bundle [com.atlassian.jira.jira-project
s-plugin] got ActiveObjectsConfiguration
2016-05-04 08:35:43,160 active-objects-init-JiraTenantImpl{id='system'}-0 DEBUG jkowall     [c.a.activeobj
ects.osgi.TenantAwareActiveObjects] bundle [com.atlassian.jira.jira-projects-plugin] creating ActiveObject
s
2016-05-04 08:35:43,215 active-objects-init-JiraTenantImpl{id='system'}-0 DEBUG jkowall     [c.a.activeobj
ects.osgi.TenantAwareActiveObjects] bundle [com.atlassian.jira.jira-projects-plugin] created ActiveObjects
2016-05-04 08:35:43,265 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.OsgiServiceUtilsImpl] Registering service net.java.ao.atlassian.A
tlassianTableNameConverter@3b7c5aba with interface net.java.ao.schema.TableNameConverter and properties {c
om.atlassian.plugin.key=com.atlassian.jira.plugins.jira-transition-triggers-plugin}
2016-05-04 08:35:43,268 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.OsgiServiceUtilsImpl] Registering service com.atlassian.activeobj
ects.config.internal.DefaultActiveObjectsConfiguration@37268d4d with interface com.atlassian.activeobjects
.config.ActiveObjectsConfiguration and properties {com.atlassian.plugin.key=com.atlassian.jira.plugins.jir
a-transition-triggers-plugin}
2016-05-04 08:35:43,271 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.ActiveObjectsServiceFactory] onPluginModuleEnabledEvent storing u
nattached <ao> configuration module for [com.atlassian.jira.plugins.jira-transition-triggers-plugin]
2016-05-04 08:35:43,284 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.ActiveObjectsServiceFactory] onPluginEnabledEvent attaching unbou
nd <ao> to [com.atlassian.jira.plugins.jira-transition-triggers-plugin]
2016-05-04 08:35:43,285 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.TenantAwareActiveObjects] init bundle [com.atlassian.jira.plugins
.jira-transition-triggers-plugin]
```

# Use Cases for Data Types

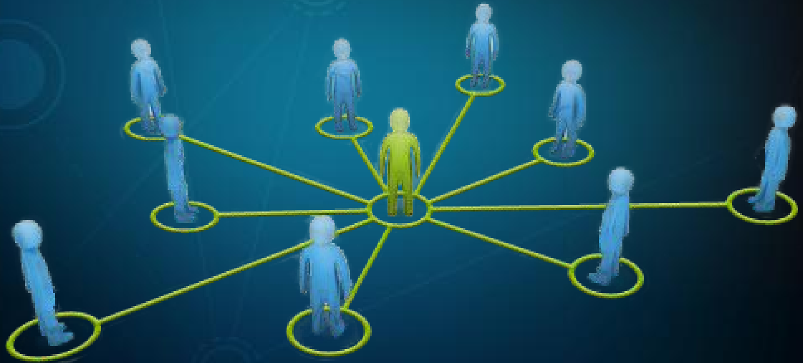
- Metrics
  - Average, Peak
  - Percentage
  - Correlation to Metrics and Events
- Events
  - Search
  - Parse
  - Correlate to Metrics



```
s-plugin] loading new AO promise for JiraTenantImpl{id='system'}
2016-05-04 08:35:43,154 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.TenantAwareActiveObjects] setAoConfiguration [com.atlassian.jira.
jira-projects-plugin]
2016-05-04 08:35:43,154 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.TenantAwareActiveObjects] bundle [com.atlassian.jira.jira-project
s-plugin] got ActiveObjectsConfiguration
2016-05-04 08:35:43,160 active-objects-init-JiraTenantImpl{id='system'}-0 DEBUG jkowall     [c.a.activeobj
ects.osgi.TenantAwareActiveObjects] bundle [com.atlassian.jira.jira-projects-plugin] creating ActiveObject
s
2016-05-04 08:35:43,215 active-objects-init-JiraTenantImpl{id='system'}-0 DEBUG jkowall     [c.a.activeobj
ects.osgi.TenantAwareActiveObjects] bundle [com.atlassian.jira.jira-projects-plugin] created ActiveObjects
2016-05-04 08:35:43,265 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.OsgiServiceUtilsImpl] Registering service net.java.ao.atlassian.A
tlassianTableNameConverter@3b7c5aba with interface net.java.ao.schema.TableNameConverter and properties {c
om.atlassian.plugin.key=com.atlassian.jira.plugins.jira-transition-triggers-plugin}
2016-05-04 08:35:43,268 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.OsgiServiceUtilsImpl] Registering service com.atlassian.activeobj
ects.config.internal.DefaultActiveObjectsConfiguration@37268d4d with interface com.atlassian.activeobjects
.config.ActiveObjectsConfiguration and properties {com.atlassian.plugin.key=com.atlassian.jira.plugins.jir
a-transition-triggers-plugin}
2016-05-04 08:35:43,271 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.ActiveObjectsServiceFactory] onPluginModuleEnabledEvent storing u
nattached <ao> configuration module for [com.atlassian.jira.plugins.jira-transition-triggers-plugin]
2016-05-04 08:35:43,284 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.ActiveObjectsServiceFactory] onPluginEnabledEvent attaching unbou
nd <ao> to [com.atlassian.jira.plugins.jira-transition-triggers-plugin]
2016-05-04 08:35:43,285 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
est/plugins/1.0/ [c.a.activeobjects.osgi.TenantAwareActiveObjects] init bundle [com.atlassian.jira.plugins
.jira-transition-triggers-plugin]
2016-05-04 08:35:43,285 UpmAsynchronousTaskManager:thread-1 DEBUG jkowall 514x258x1 k17m7f 192.168.0.10 /r
```

# Software Instrumentation

- Logging
  - Supplied by vendor
  - Created by developers
  - Not easily controlled



- Push Collection
  - Attach and extract
  - Software agent or network tap
- Pull Collection
  - Polling APIs - HTTP, SNMP, WMI

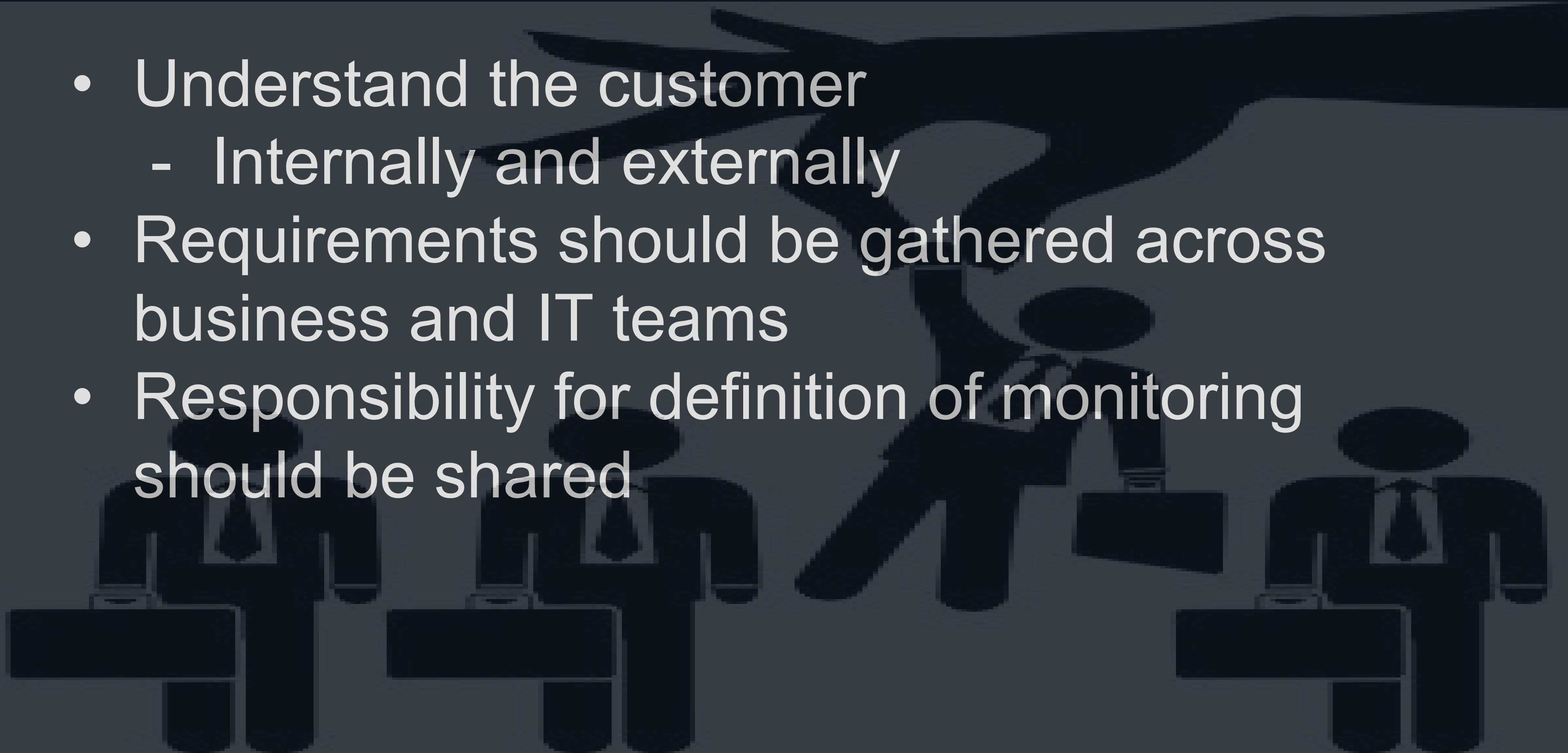
# Priorities for Instrumentation

- Business
- Application
- Services
- Infrastructure

## Events and Metrics

# Business Instrumentation

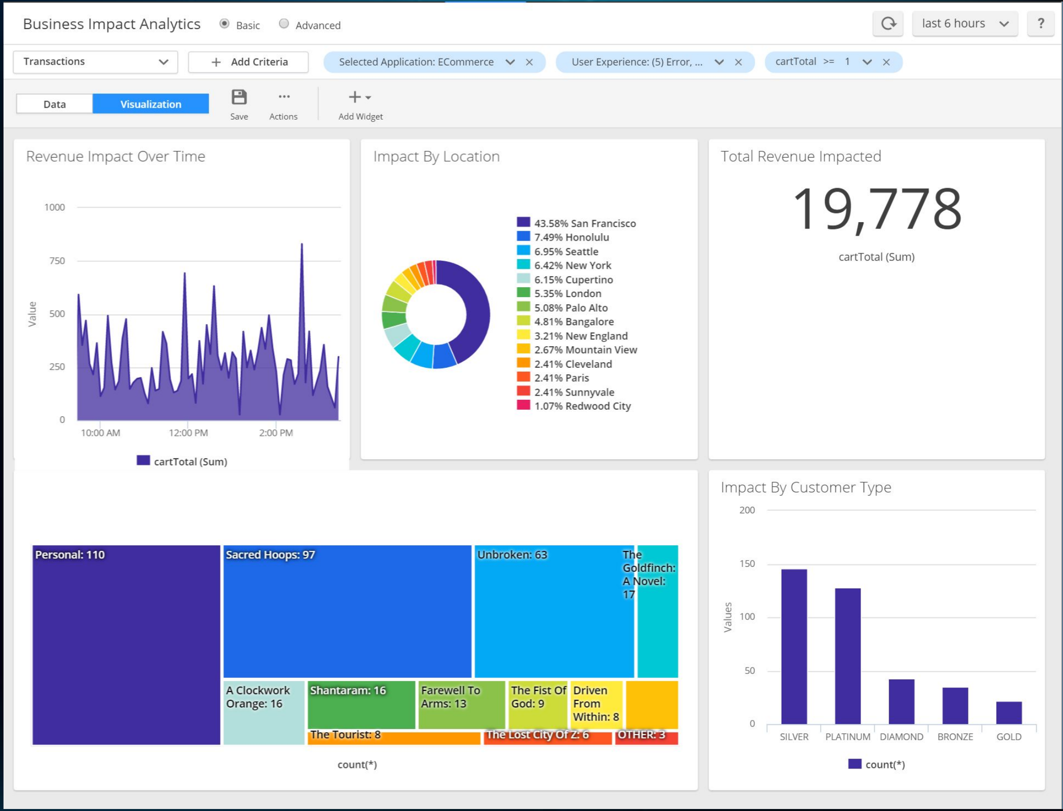
# Uplevel the Conversation



- Understand the customer
  - Internally and externally
- Requirements should be gathered across business and IT teams
- Responsibility for definition of monitoring should be shared

# Business Metrics and KPIs

- Customer Metrics
  - Conversion between products
  - Loyalty and retention (churn)
  - Usage metrics (feature and product)
- Sales / Marketing Metrics
  - Revenue
  - Cost of customer acquisition
  - User flows through applications



Business Impact Analytics ● Basic ○ Advanced

last 6 hours

Transactions | Add Criteria | Selected Application: ECommerce | User Experience: (5) Error, ... | cartTotal >= 1

# Technical Metrics and KPIs

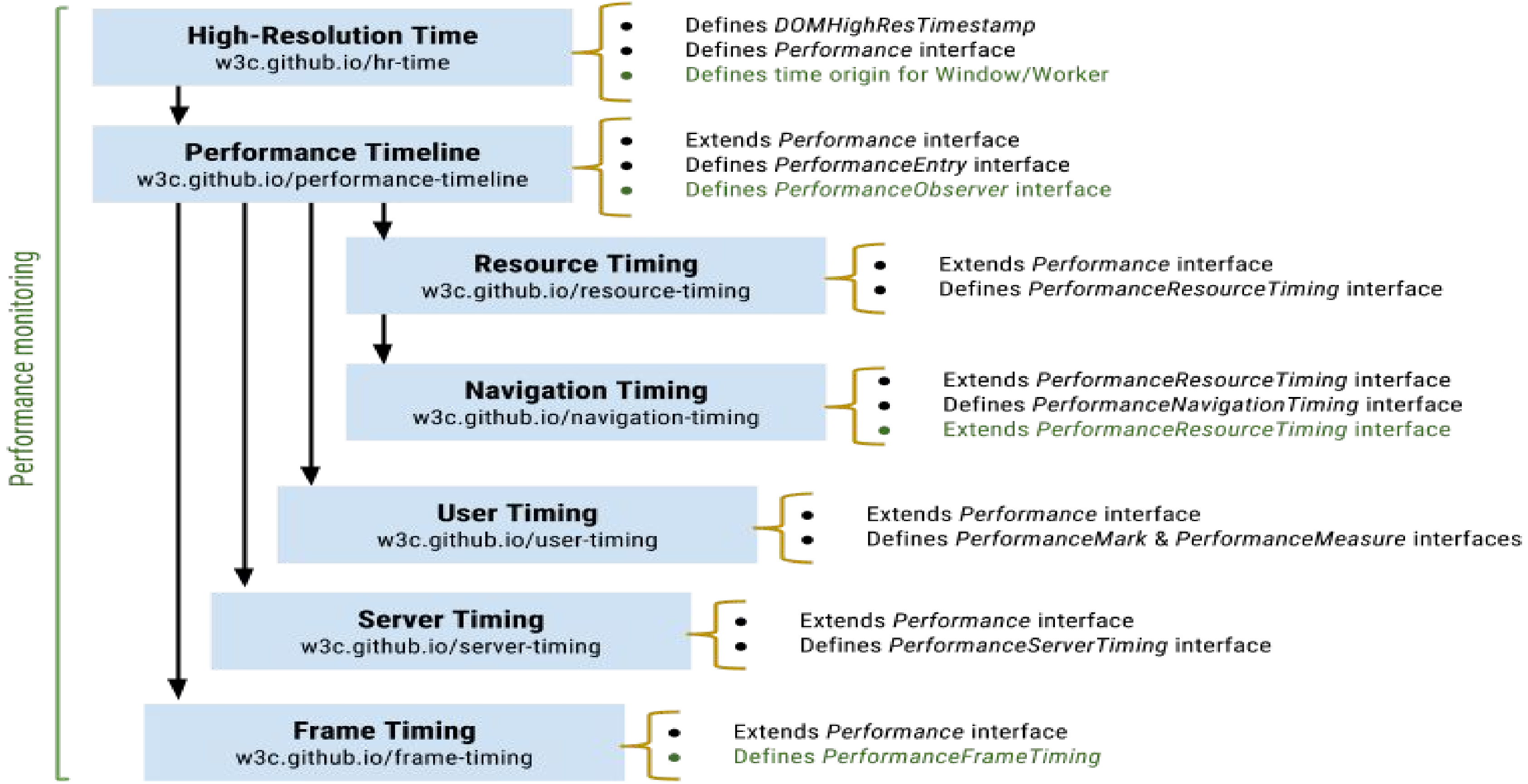
- End to end performance
  - User through transaction hops
  - Error isolation
- End user experience
  - Client side errors
  - Latency per element (page or app) + 3rd party
  - Client side DNS
- Application component performance
  - Metrics from app server
  - Metrics from code
  - Queries
  - Errors
- Intra application component performance

# Use Cases for Business and Technical Data

- Usage
- Problem identification - MTTI
- Problem resolution - MTTR
- User satisfaction
- Usability
- Performance
- Change analysis
    - A/B testing
    - data center moves
    - technology changes

# Frontend Instrumentation

# Instrumenting Browsers



Performance monitoring

**High-Resolution Time**
w3c.github.io/hr-time

- Defines *DOMHighResTimestamp*
- Defines *Performance* interface
- Defines time origin for Window/Worker

**Performance Timeline**
w3c.github.io/performance-timeline

- Extends *Performance* interface
- Defines *PerformanceEntry* interface
- Defines *PerformanceObserver* interface

**Resource Timing**
w3c.github.io/resource-timing

- Extends *Performance* interface
- Defines *PerformanceResourceTiming* interface

**Navigation Timing**
w3c.github.io/navigation-timing

- Extends *PerformanceResourceTiming* interface
- Defines *PerformanceNavigationTiming* interface
- Extends *PerformanceResourceTiming* interface

**User Timing**
w3c.github.io/user-timing

- Extends *Performance* interface
- Defines *PerformanceMark* & *PerformanceMeasure* interfaces

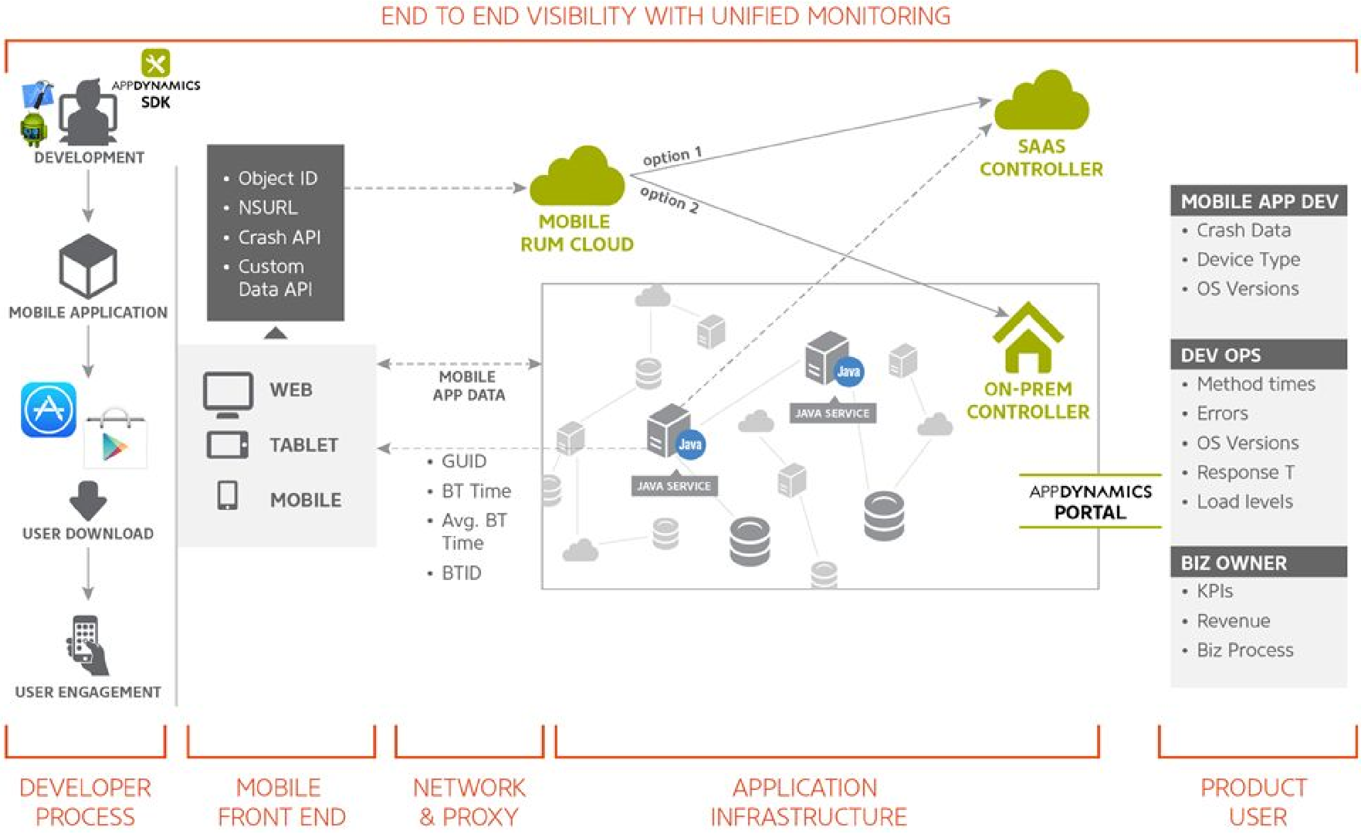
**Server Timing**
w3c.github.io/server-timing

- Extends *Performance* interface
- Defines *PerformanceServerTiming* interface

**Frame Timing**
w3c.github.io/frame-timing

- Extends *Performance* interface
- Defines *PerformanceFrameTiming*

# Instrumenting Mobile



END TO END VISIBILITY WITH UNIFIED MONITORING

APPDYNAMICS SDK

DEVELOPMENT

MOBILE APPLICATION

USER DOWNLOAD

USER ENGAGEMENT

- Object ID
- NSURL
- Crash API
- Custom Data API

WEB

TABLET

MOBILE

MOBILE APP DATA

- GUID
- BT Time
- Avg. BT Time
- BTID

MOBILE RUM CLOUD

option 1

option 2

SAAS CONTROLLER

ON-PREM CONTROLLER

JAVA SERVICE

Java

JAVA SERVICE

Java

APPDYNAMICS PORTAL

MOBILE APP DEV

- Crash Data
- Device Type
- OS Versions

DEV OPS

- Method times
- Errors
- OS Versions
- Response T
- Load levels

BIZ OWNER

- KPIs
- Revenue
- Biz Process

DEVELOPER PROCESS

MOBILE FRONT END

NETWORK & PROXY

APPLICATION INFRASTRUCTURE

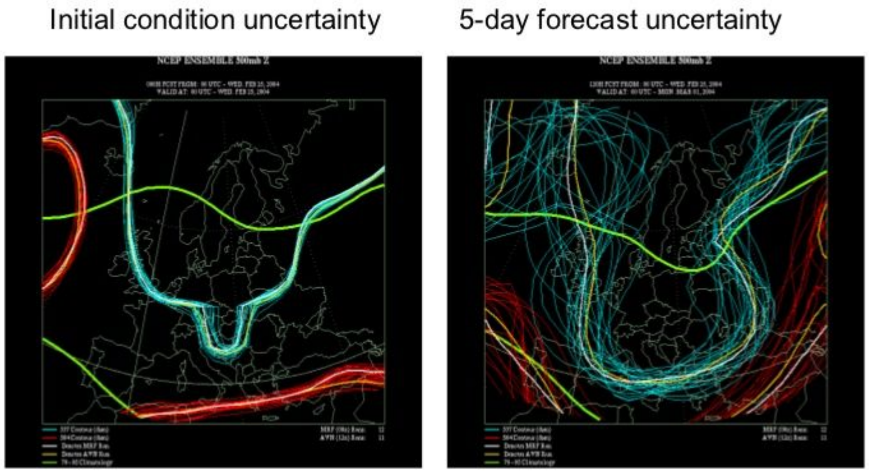
PRODUCT USER

# Simulating Users

Synthetic transactions are good for

- SLAs
- Availability
- Baseline performance
- DNS
- SSL

But if you try to use it as a barometer you will fail

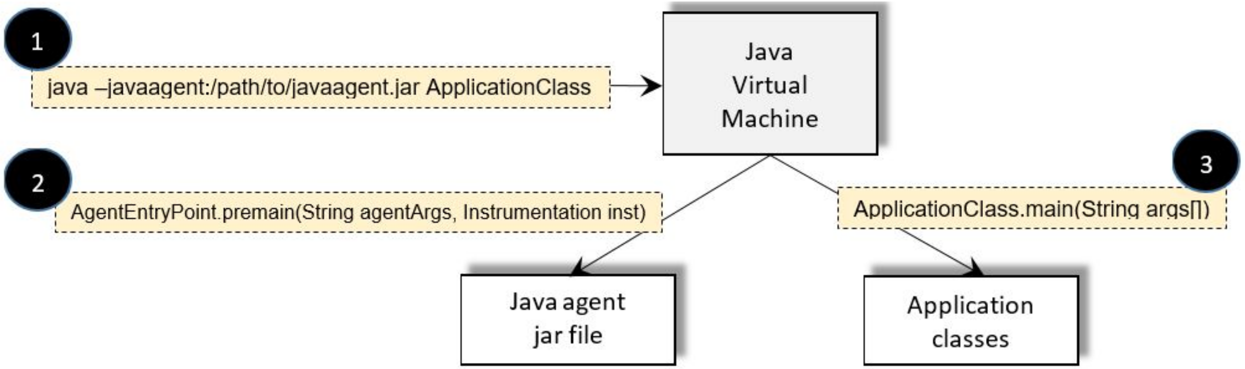


Initial condition uncertainty

5-day forecast uncertainty

# Backend Instrumentation of Java and .NET

# Java Instrumentation



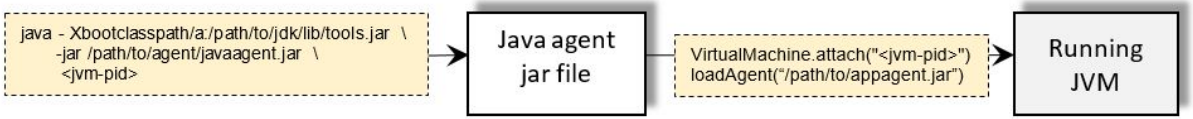
```
java –javaagent:/path/to/javaagent.jar ApplicationClass
```

```
AgentEntryPoint.premain(String agentArgs, Instrumentation inst)
```

```
ApplicationClass.main(String args[])
```

JSR-163 (Java$^{TM}$ Platform Profiling Architecture) added in Java 1.5

Overloads the default behavior of Java to allow hooks into code for many use cases

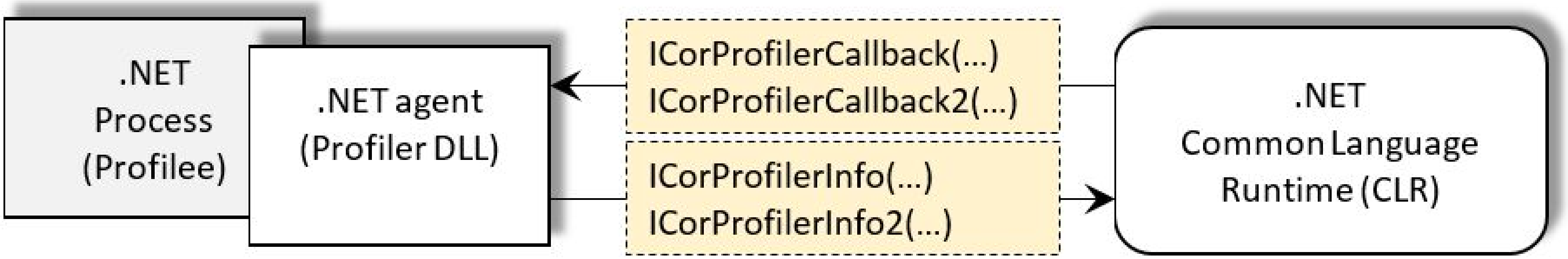


```
java - Xbootclasspath/a:/path/to/jdk/lib/tools.jar  \
     -jar /path/to/agent/javaagent.jar  \
      <jvm-pid>
```

```
VirtualMachine.attach("<jvm-pid>")
loadAgent("/path/to/appagent.jar")
```

Since JDK 1.6, for the Oracle HotSpot JVM, a javaagent may be dynamically attached to a running JVM by specifying the process-id (pid).

# .NET Instrumentation



- Profiling API loaded into the same process as the application process that is being profiled.
- Callback interface (ICorProfilerCallback in the .NET Framework version 1.0 and 1.1, ICorProfilerCallback2 in version 2.0 and later)
- CLR calls the methods in that interface to notify the .NET agent of events in the profiled process
- Profiler can also call into the CLR by using the methods in the ICorProfilerInfo and ICorProfilerInfo2 interfaces to obtain information about the state of the profiled application
- Callbacks are used to inject MSIL (Microsoft Intermediate Language) bytecode into existing application code for instrumentation.

# Backend Instrumentation of Interpreted Languages

# Monkey Patching

Wikipedia relevant definition:

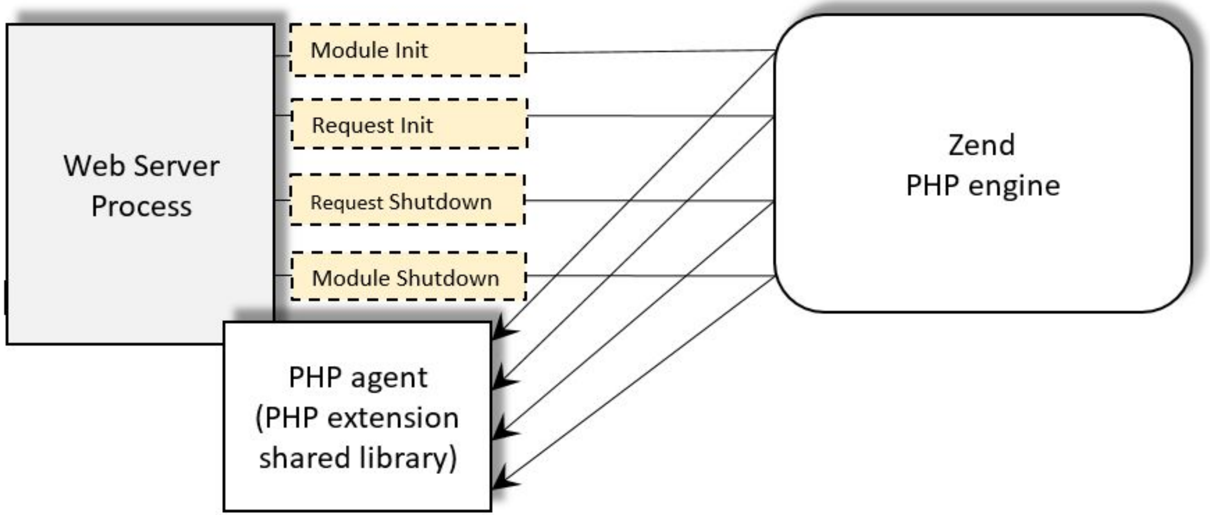
In Ruby,[3] Python,[4] and many other dynamic programming languages... dynamic modifications of a class or module at runtime, motivated by the intent to patch existing third-party code as a workaround to a bug or feature which does not act as desired

- Replace methods / attributes / functions at runtime
- Apply a patch at runtime to the objects in memory, instead of the source code on disk;

***Disclaimer : Can be very dangerous, hard to maintain***

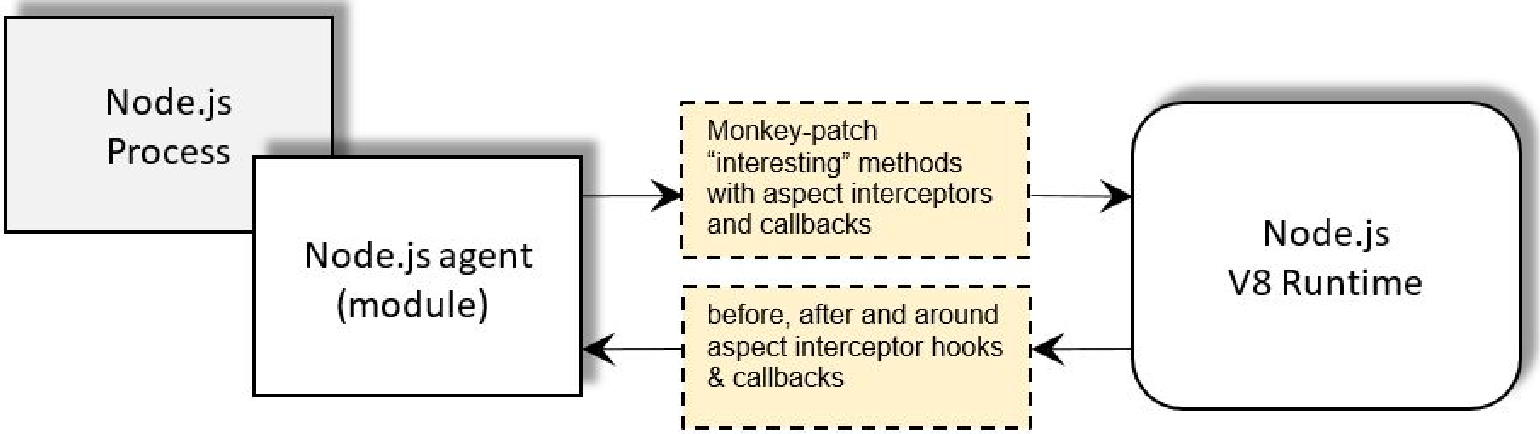
# PHP Instrumentation

Handles state changes and new web server initialization (which are PHP instances)



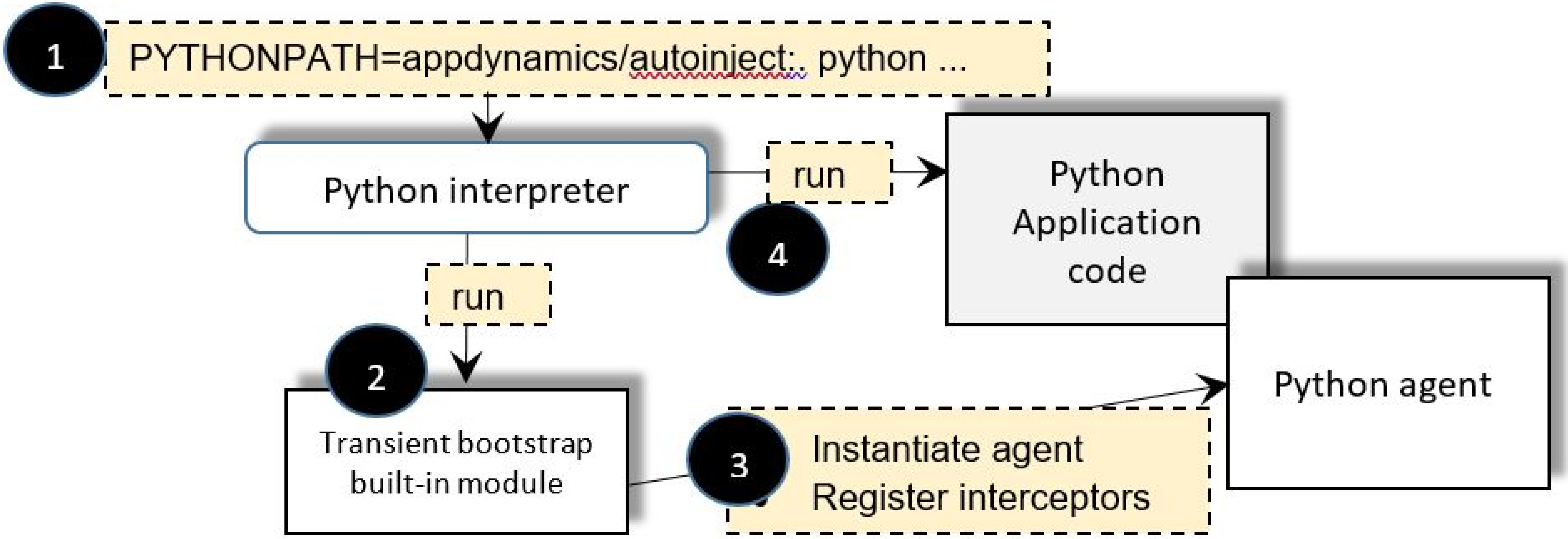
Zend callback methods *zend_execute(…)*, *zend_execute_internal(…)* and *zend_compile_file(…)* so that it can wrap the original implementations with instrumentation code.

# Node.js Instrumentation


Node.js Process
Node.js agent (module)
Monkey-patch "interesting" methods with aspect interceptors and callbacks
before, after and around aspect interceptor hooks & callbacks
Node.js V8 Runtime

- Wrap methods using before, after and around aspect interceptors.
- Callback along with after, before and around aspect interceptor.
- Notifications when asynchronous calls are complete.

# Python Instrumentation

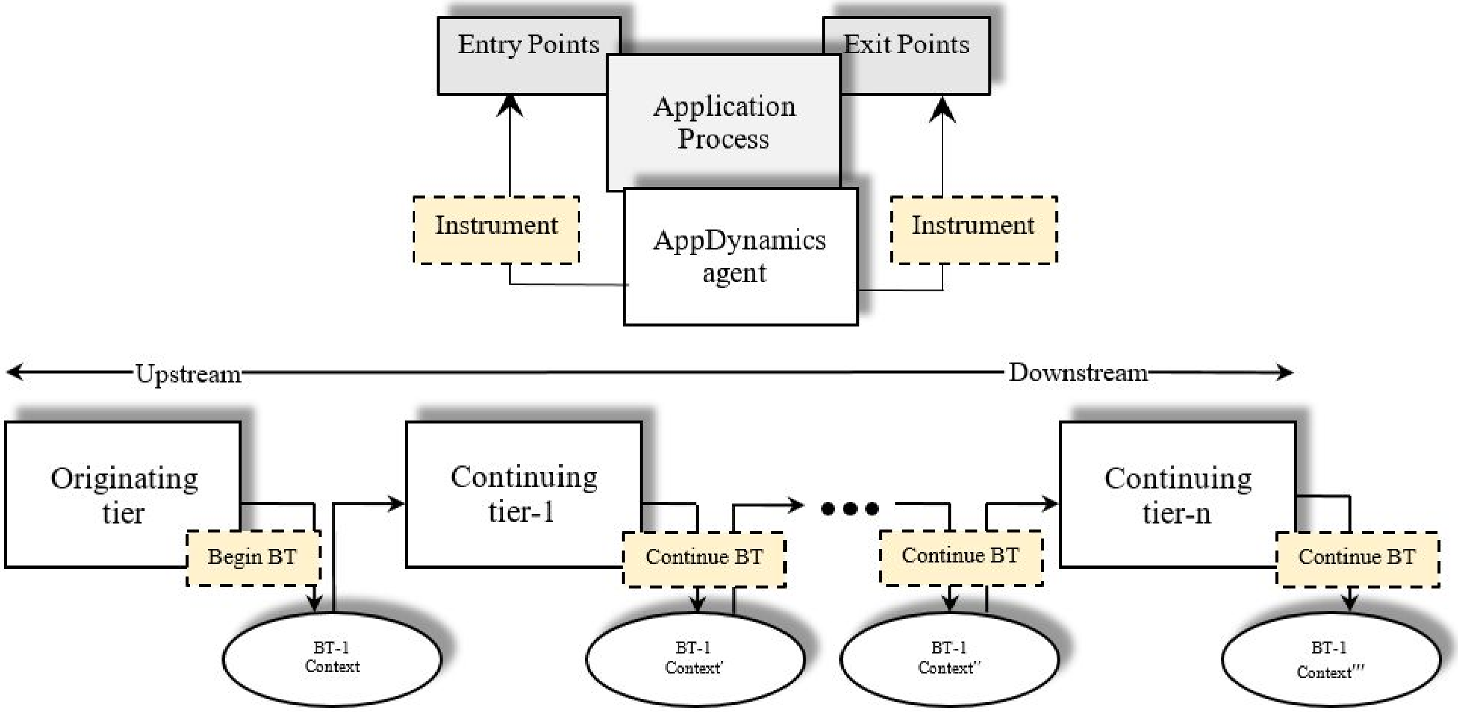


# Logging Best Practices

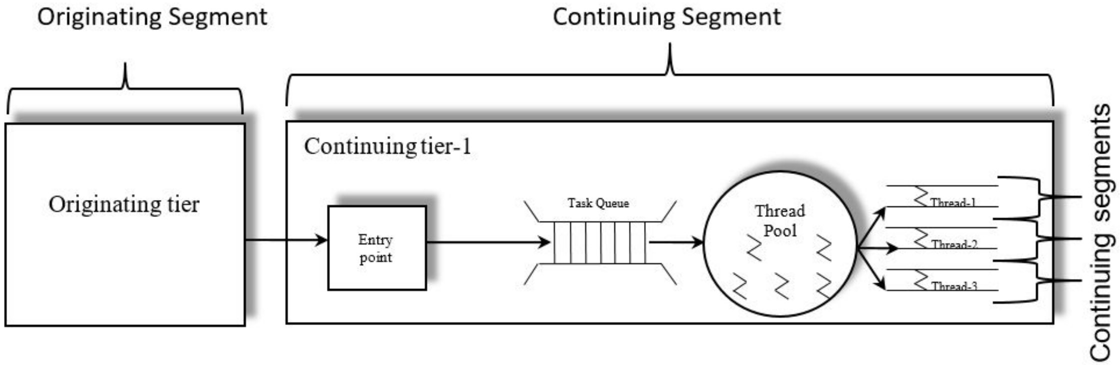
- Easily parsed (JSON)
- Time (long), Source
- Log errors and exceptions
- Logs are not transaction records, they are not good metric stores
- Write your own identifiers for each statement logged (or instrument and inject)
- Think about security implications (plain text, on disk, syslog insecure)
- Keep small (thanks Java, .NET…)
- Don’t overdo it (performance implications)

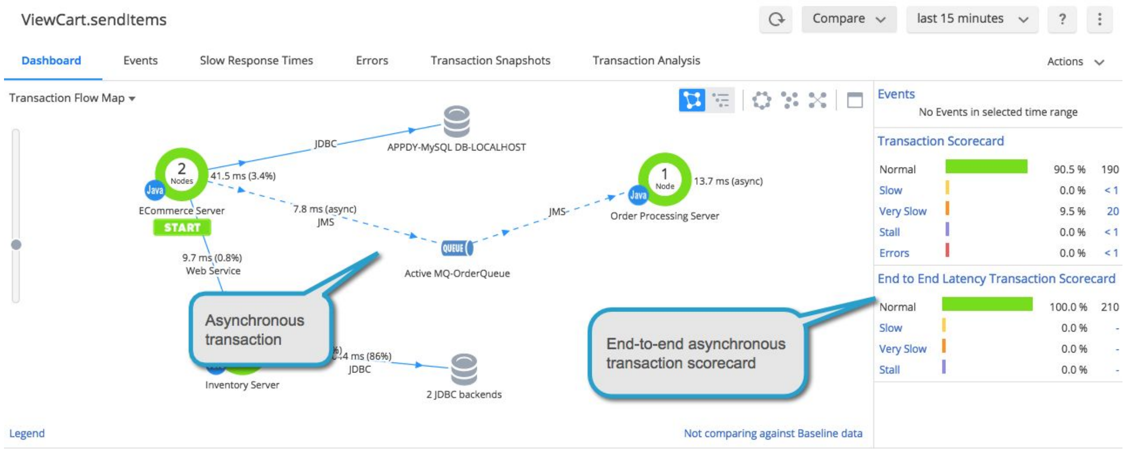
# Transaction Correlation

# Correlation in end to end APM



# Correlation in asynchronous calls (headache)





Events

No Events in selected time range

Transaction Scorecard

| | | | |
|---|---|---|---|
| Normal | 🟩 | 90.5 % | 190 |
| Slow | | 0.0 % | < 1 |
| Very Slow | | 9.5 % | 20 |
| Stall | | 0.0 % | < 1 |
| Errors | | 0.0 % | < 1 |

End to End Latency Transaction Scorecard

| | | | |
|---|---|---|---|
| Normal | 🟩 | 100.0 % | 210 |
| Slow | | 0.0 % | - |
| Very Slow | | 0.0 % | - |
| Stall | | 0.0 % | - |

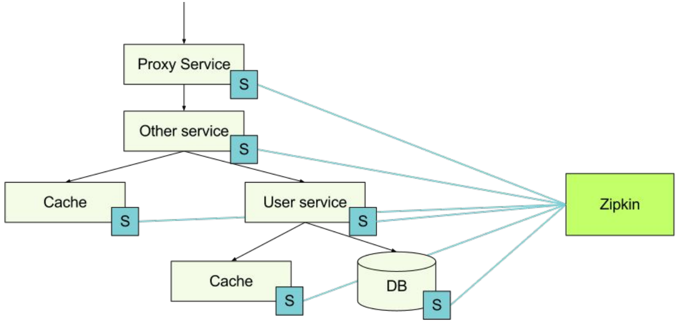
Asynchronous transaction

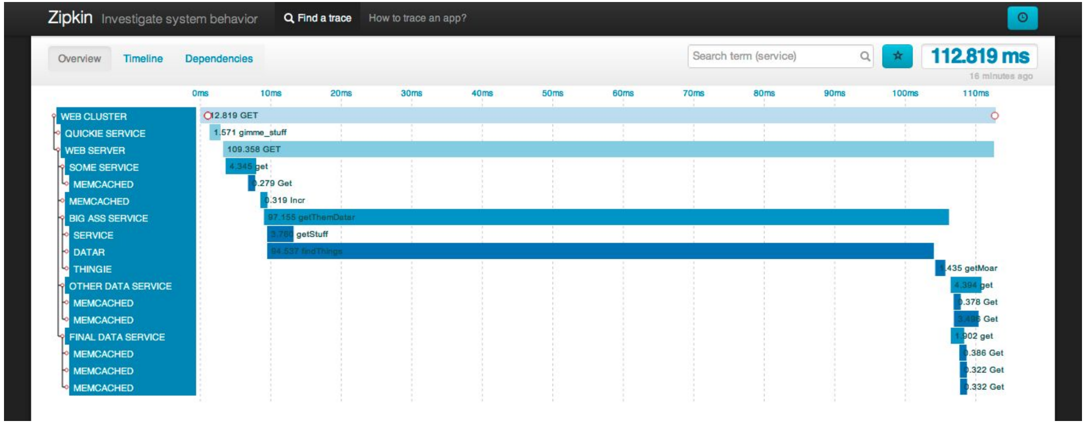
End-to-end asynchronous transaction scorecard

# Correlation in Open Source

Zipkin is the most advanced, many new forks and instrumentations (Java, JavaScript, Python, Ruby, C#, Go)

No async support :(



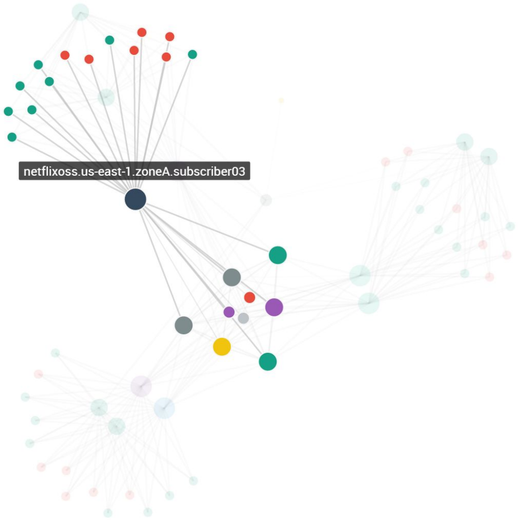


| | |
|---|---|
| WEB CLUSTER | 112.819 GET |
| QUICKIE SERVICE | 1.571 gimme_stuff |
| WEB SERVER | 109.358 GET |
| SOME SERVICE | 4.345 get |
| MEMCACHED | 0.279 Get |
| MEMCACHED | 0.319 Incr |
| BIG ASS SERVICE | 97.155 getThemDatar |
| SERVICE | 3.760 getStuff |
| DATAR | 94.537 findThings |
| THINGIE | 1.435 getMoar |
| OTHER DATA SERVICE | 4.394 get |
| MEMCACHED | 0.378 Get |
| MEMCACHED | 3.496 Get |
| FINAL DATA SERVICE | 1.902 get |
| MEMCACHED | 0.386 Get |
| MEMCACHED | 0.322 Get |
| MEMCACHED | 0.332 Get |

# Future of Correlation in Open Source

- OpenTracing an open standard API for instrumentation
    - Doesn’t manage overhead, can hang yourself
- PivotTracing runs distributed traces on demand
    - After issues detected

- Spigo visualization and simulation lots of evolution



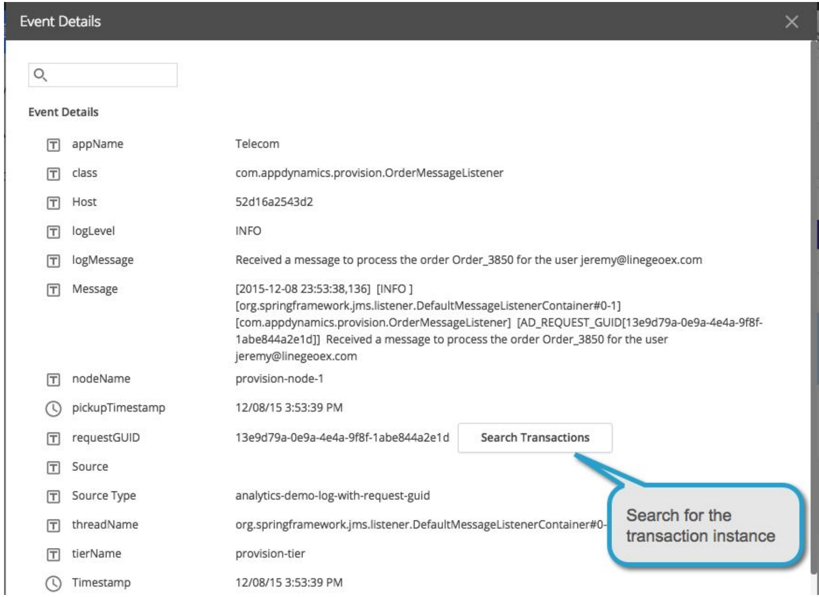
# Correlation in Logs

- Log every transaction segment
- Persist a GUID or transaction ID
- This is very difficult in large teams
- Inefficient to analyze and pull metrics from logs
- Doesn’t work unless you own the code

```
[code]
PERF,2013-04-03 11:29:52.640,external,0x123456,NA,service1,MyAPP,jimmy,NA,336,NA,NA
INFO,2013-04-03 11:29:53.189,internal,789012,0x123456,service2,TheirApp,jimmy,NA,174,NA,NA
INFO,2013-04-03 11:29:52.892,internal,345678,789012,service3,TheirApp,jimmy,NA,163,NA,NA
[/code]
```

# Transaction Correlation and Logs!

- Many integrations across APM and Log vendors
- Can add correlation in code and use any log tool
  - ex: [%X{AD.requestGUID}]
- We auto inject and correlate (one platform)



Event Details

Event Details

| | |
|---|---|
| appName | Telecom |
| class | com.appdynamics.provision.OrderMessageListener |
| Host | 52d16a2543d2 |
| logLevel | INFO |
| logMessage | Received a message to process the order Order_3850 for the user jeremy@linegeoex.com |
| Message | [2015-12-08 23:53:38,136] [INFO ] [org.springframework.jms.listener.DefaultMessageListenerContainer#0-1] [com.appdynamics.provision.OrderMessageListener] [AD_REQUEST_GUID[13e9d79a-0e9a-4e4a-9f8f-1abe844a2e1d]] Received a message to process the order Order_3850 for the user jeremy@linegeoex.com |
| nodeName | provision-node-1 |
| pickupTimestamp | 12/08/15 3:53:39 PM |
| requestGUID | 13e9d79a-0e9a-4e4a-9f8f-1abe844a2e1d |
| Source | |
| Source Type | analytics-demo-log-with-request-guid |
| threadName | org.springframework.jms.listener.DefaultMessageListenerContainer#0- |
| tierName | provision-tier |
| Timestamp | 12/08/15 3:53:39 PM |

Search for the transaction instance

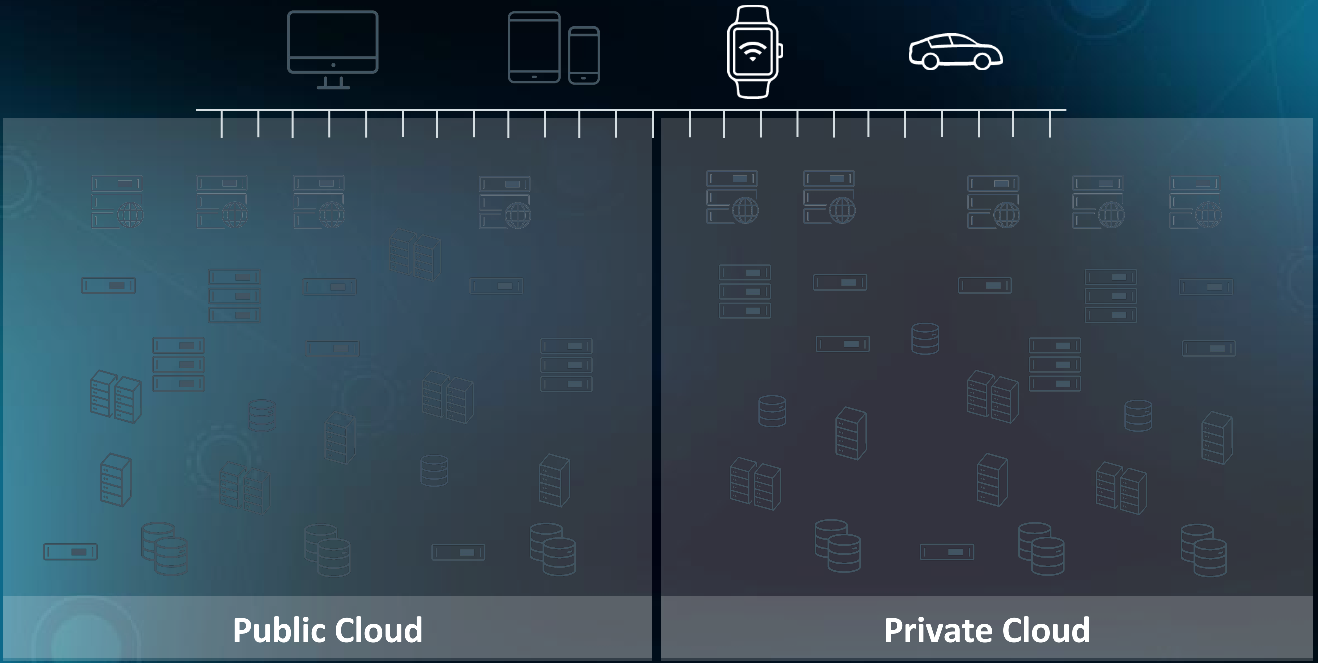
# Intro to AppDynamics

# Principles of Digital Transformation

1. The App is the Business
2. Velocity is Critical
3. Agile Architectures
4. Public Cloud

# Application Complexity is Exploding

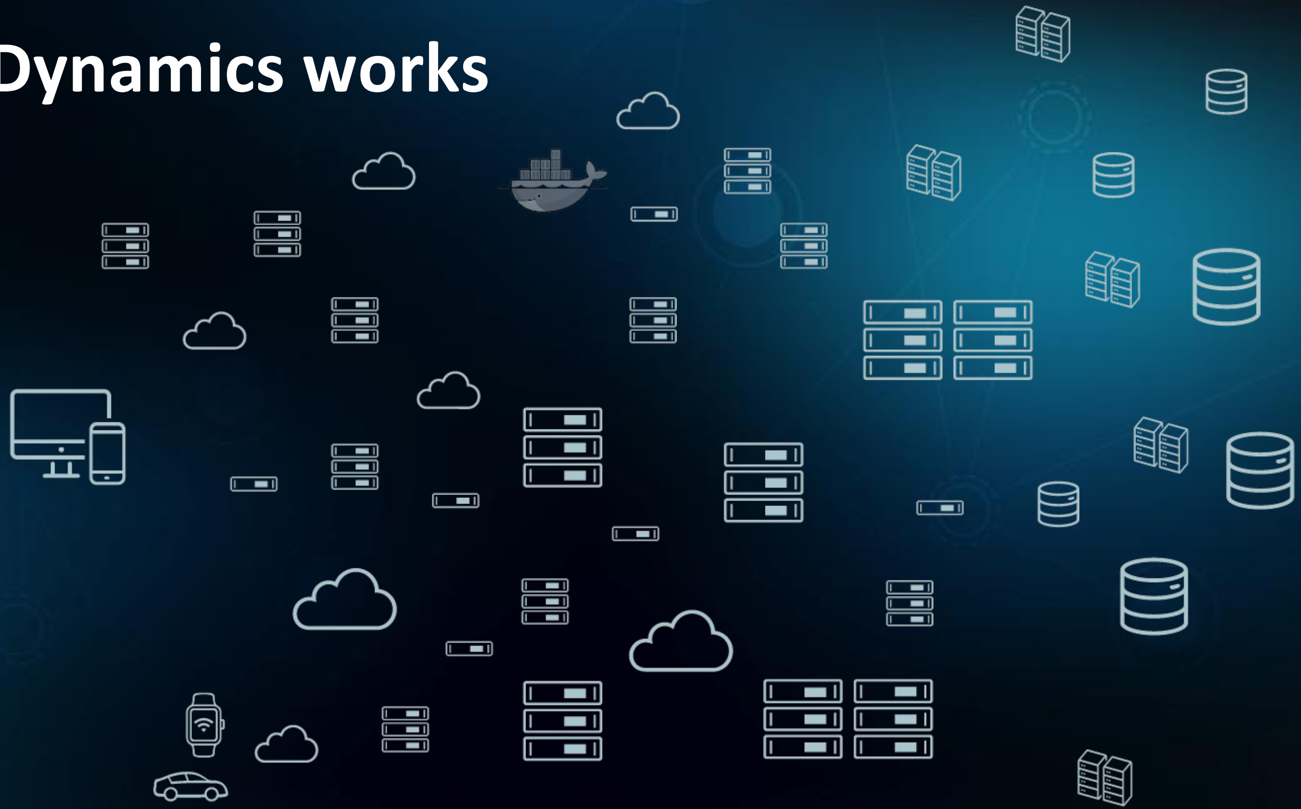
## Move Fast and *Don't* Break Things



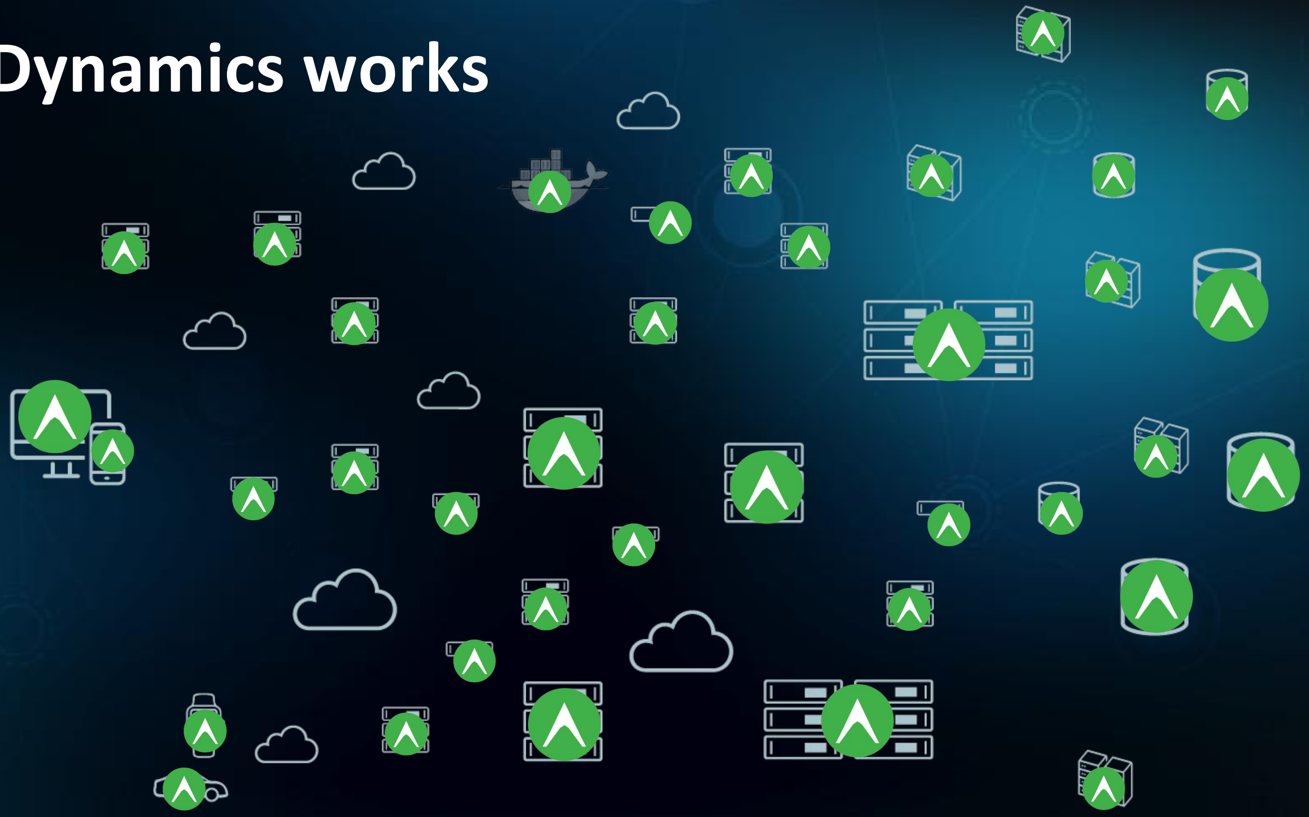
**Trends**

1. Multi-device & IoT
2. Microservices & Containers
3. Micro-databases & NoSQL
4. Hybrid Cloud
5. Continuous Deployment

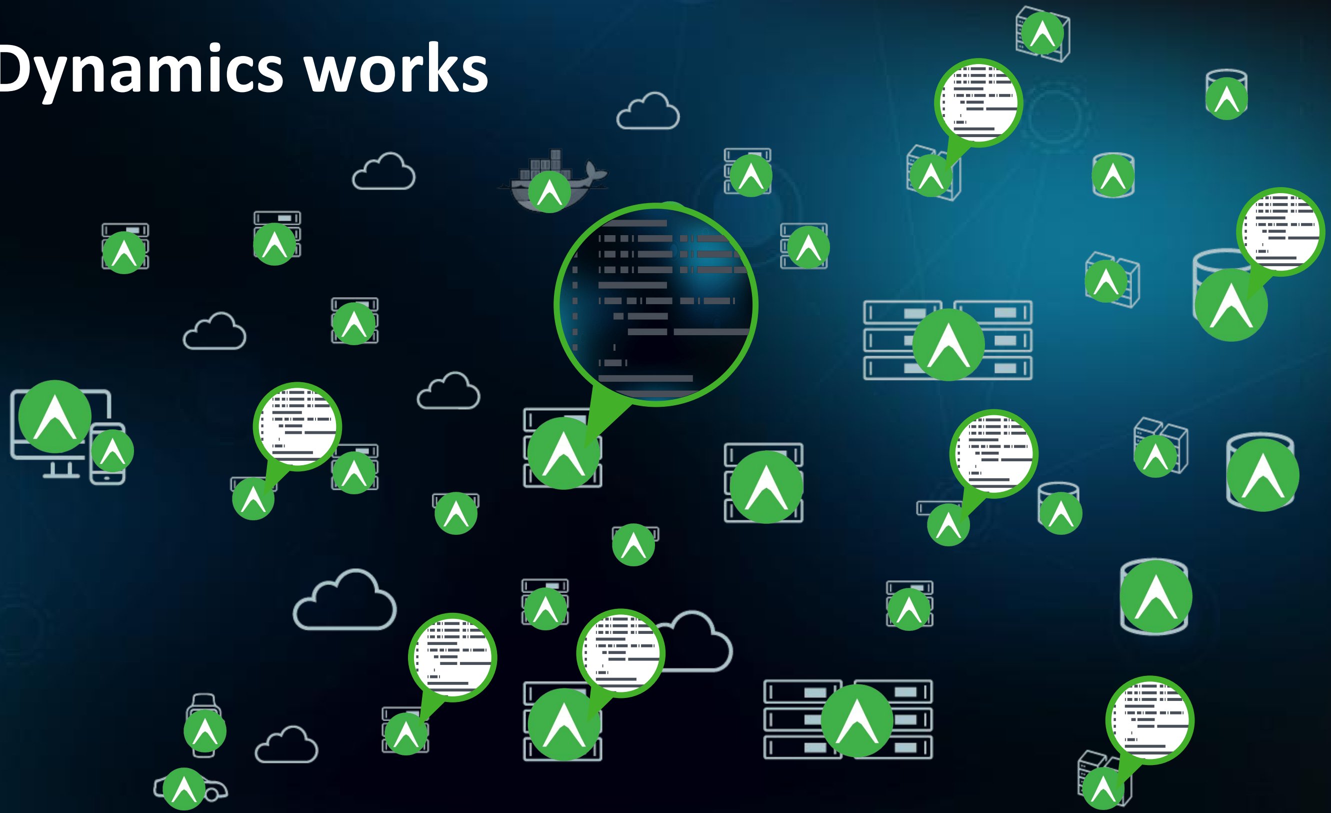
# How AppDynamics works



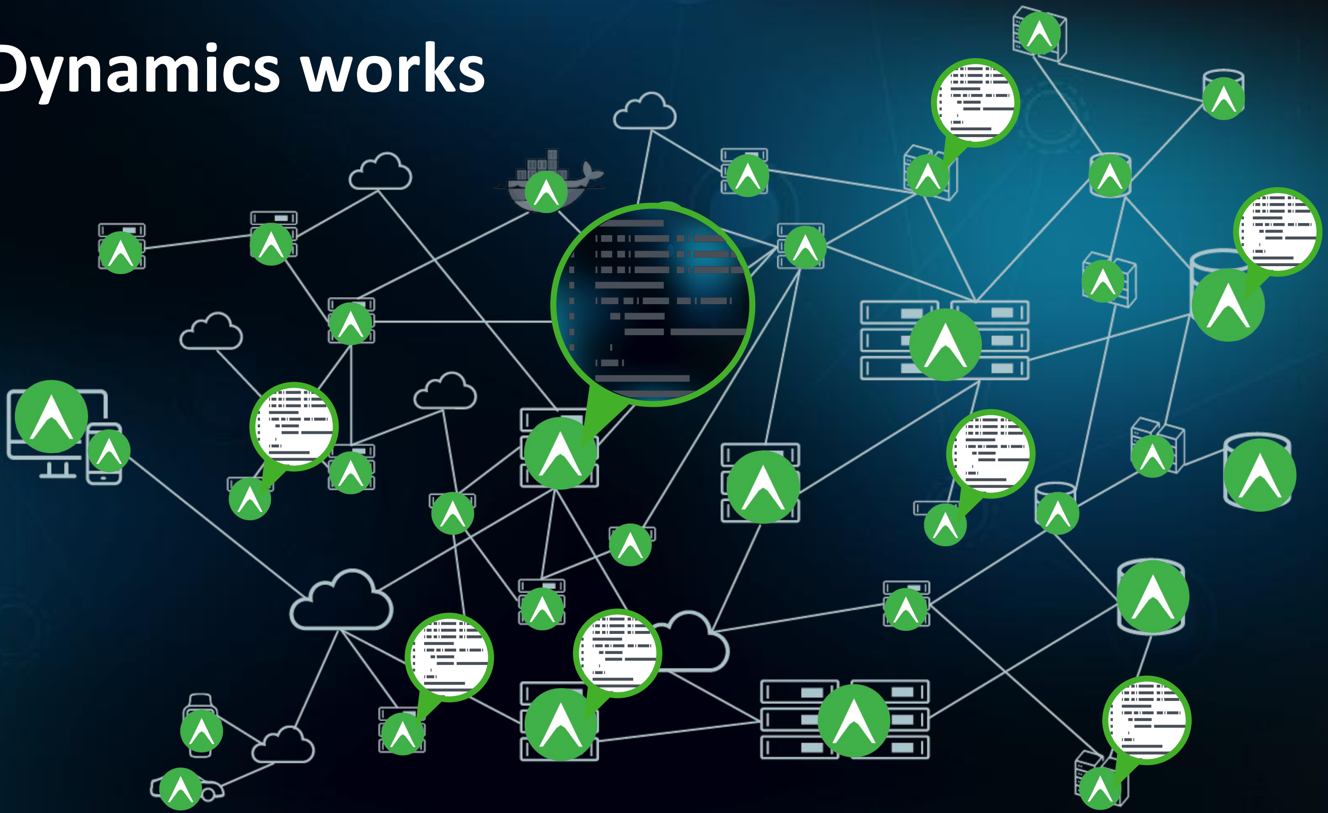
# How AppDynamics works



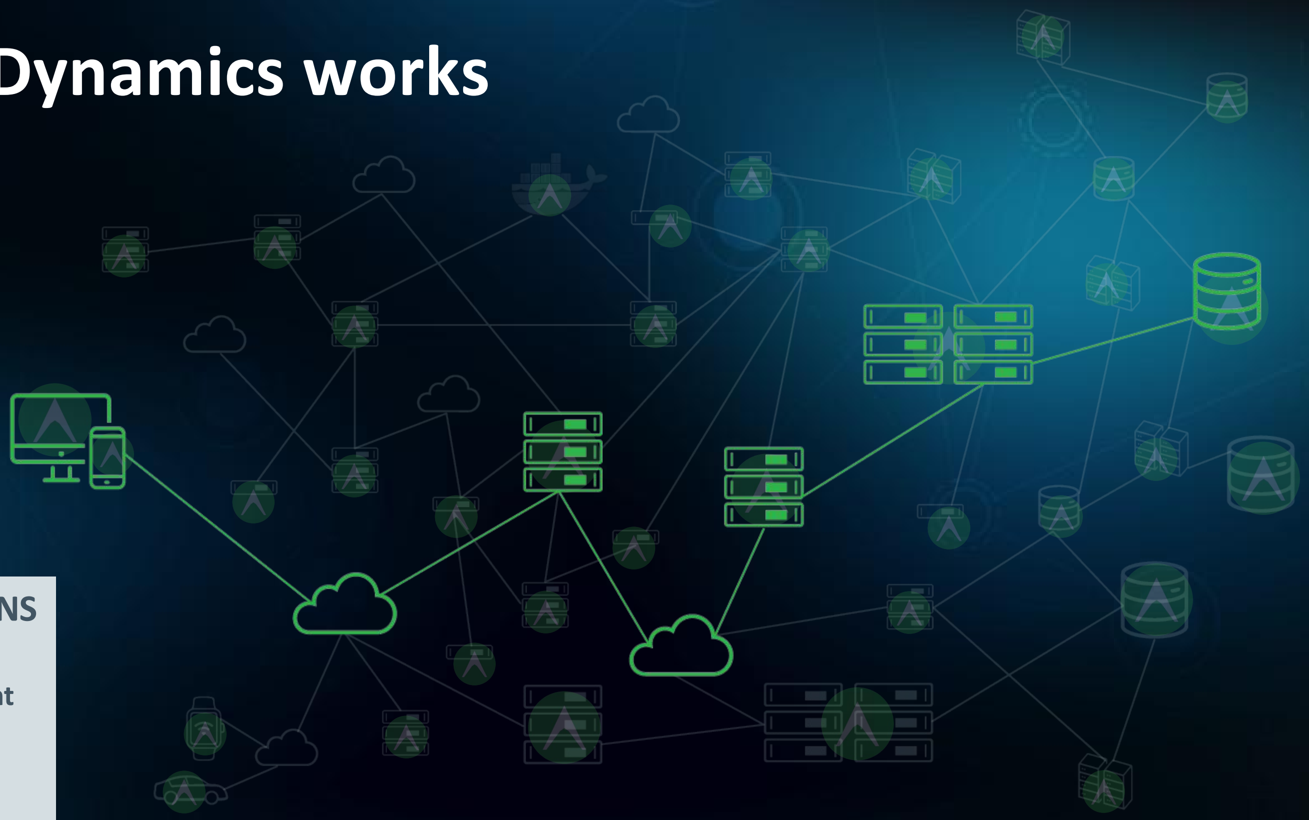
# How AppDynamics works



# How AppDynamics works



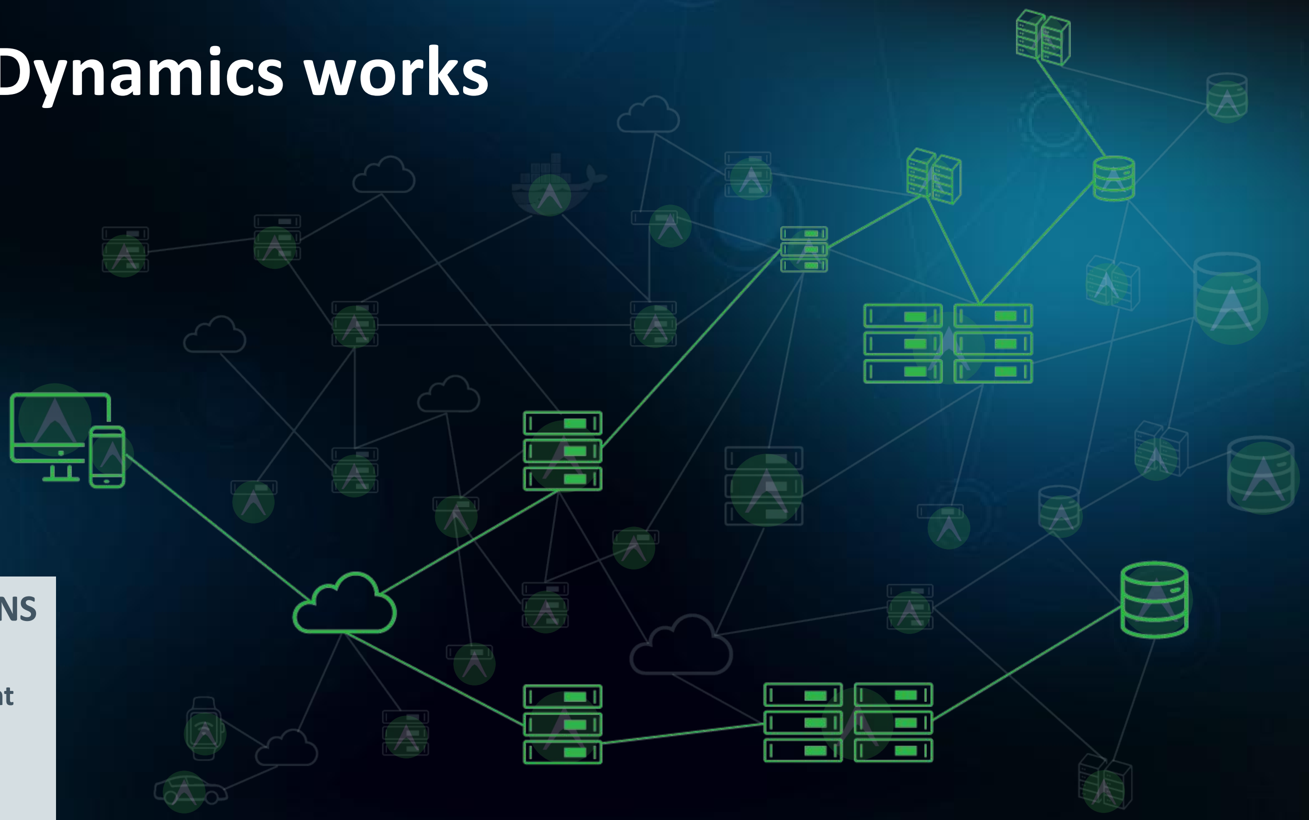
# How AppDynamics works



TRANSACTIONS

- ✓ Login
- ● Search Flight
- ● Status
- ● Purchase

# How AppDynamics works



**TRANSACTIONS**

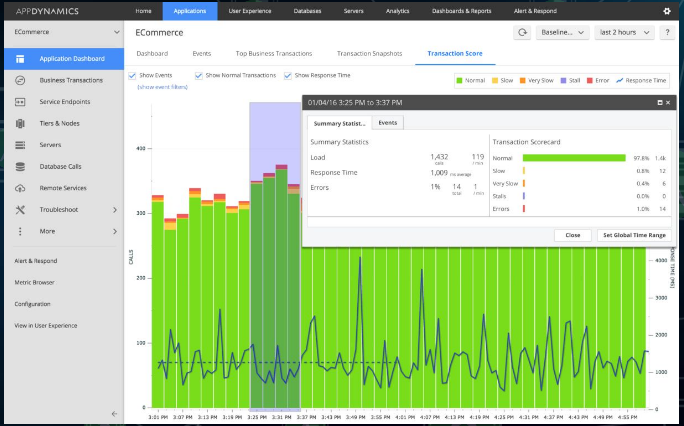
- ✓ Login
- ● Search Flight
- ● Status
- ✓ Purchase

# How AppDynamics works

Operations

## TRANSACTIONS

- ✓ Login
- ● Search Flight
- ● Status
- ● Purchase



☑ Show Events ☑ Show Normal Transactions ☑ Show Response Time
(show event filters)

01/04/16 3:25 PM to 3:37 PM

Summary Statistics

| | | |
|---|---|---|
| Load | 1,432 calls | 119 / min |
| Response Time | 1,009 ms average | |
| Errors | 1% 14 total | 1 / min |

Transaction Scorecard

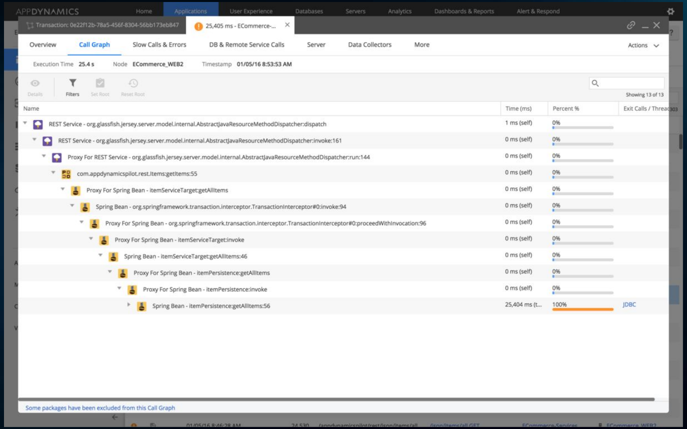
| | | |
|---|---|---|
| Normal | 97.8% | 1.4k |
| Slow | 0.8% | 12 |
| Very Slow | 0.4% | 6 |
| Stalls | 0.0% | 0 |
| Errors | 1.0% | 14 |

# How AppDynamics works

Developer

**TRANSACTIONS**

- ✓ Login
- ● Search Flight
- ● Status
- ● Purchase



Transaction: 0e22f12b-78a5-456f-8304-56bb173eb847

25,405 ms - ECommerce-...

Execution Time 25.4 s  Node ECommerce_WEB2  Timestamp 01/05/16 8:53:53 AM

Showing 13 of 13

| Name | Time (ms) | Percent % | Exit Calls / Threac |
|---|---|---|---|
| REST Service - org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher:dispatch | 1 ms (self) | 0% | |
| REST Service - org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher:invoke:161 | 0 ms (self) | 0% | |
| Proxy For REST Service - org.glassfish.jersey.server.model.internal.AbstractJavaResourceMethodDispatcher:run:144 | 0 ms (self) | 0% | |
| com.appdynamicspilot.rest.Items:getItems:55 | 0 ms (self) | 0% | |
| Proxy For Spring Bean - itemServiceTarget:getAllItems | 0 ms (self) | 0% | |
| Spring Bean - org.springframework.transaction.interceptor.TransactionInterceptor#0:invoke:94 | 0 ms (self) | 0% | |
| Proxy For Spring Bean - org.springframework.transaction.interceptor.TransactionInterceptor#0:proceedWithInvocation:96 | 0 ms (self) | 0% | |
| Proxy For Spring Bean - itemServiceTarget:invoke | 0 ms (self) | 0% | |
| Spring Bean - itemServiceTarget:getAllItems:46 | 0 ms (self) | 0% | |
| Proxy For Spring Bean - itemPersistence:getAllItems | 0 ms (self) | 0% | |
| Proxy For Spring Bean - itemPersistence:invoke | 0 ms (self) | 0% | |
| Spring Bean - itemPersistence:getAllItems:56 | 25,404 ms (t... | 100% | JDBC |

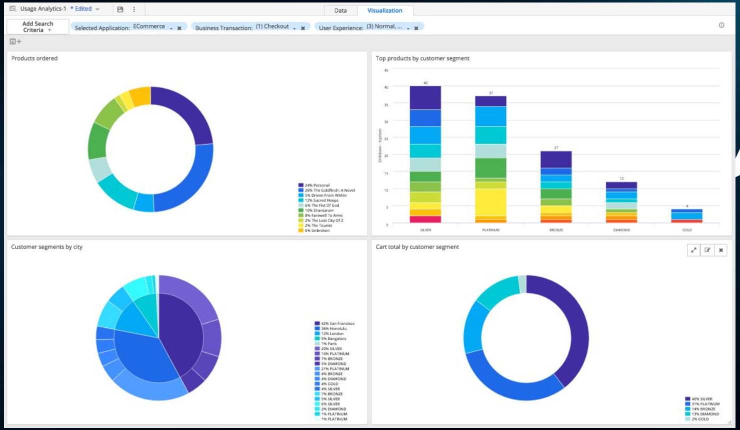
Some packages have been excluded from this Call Graph

# How AppDynamics works

Business

**TRANSACTIONS**

- ✓ Login
- ● Search Flight
- ● Status
- ● Purchase



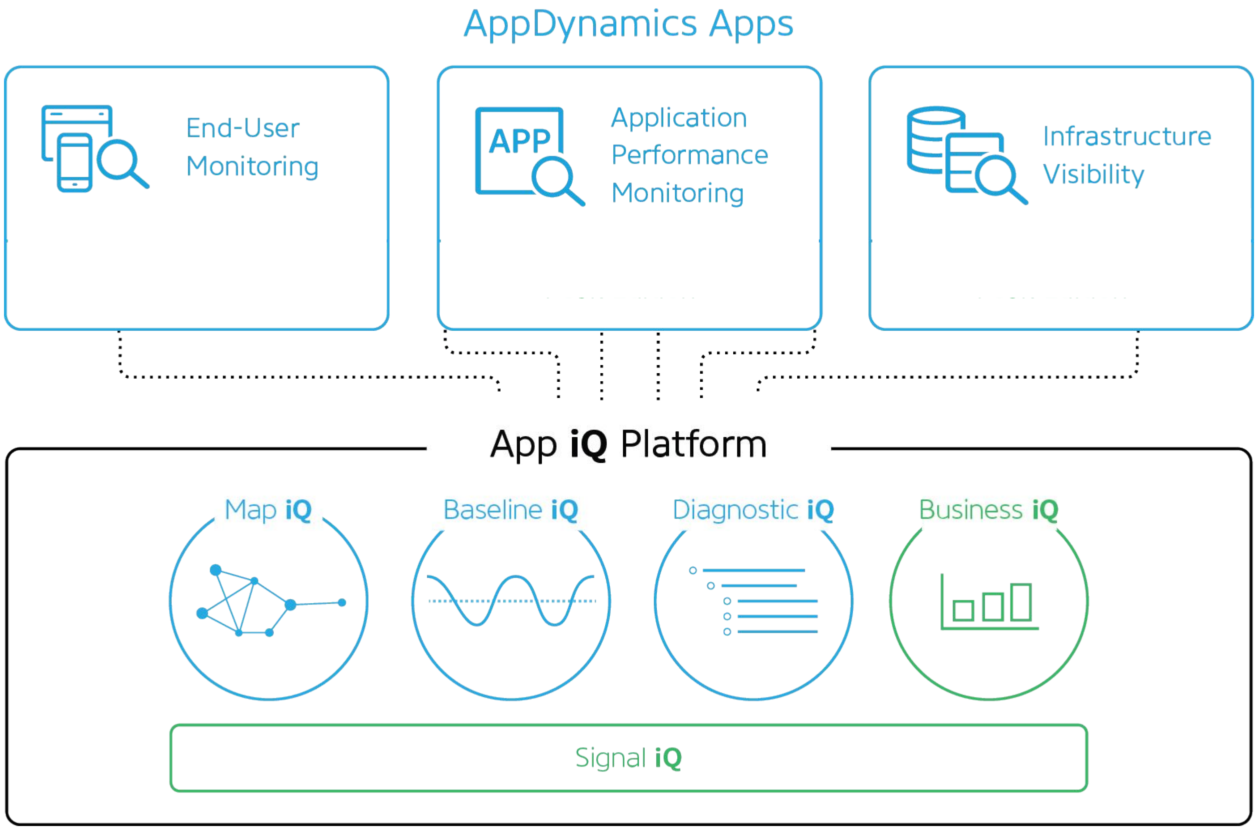
Usage Analytics-1 * Edited

Selected Application: ECommerce

Business Transaction: (1) Checkout

User Experience: (3) Normal, ...

# Application and Business Monitoring Products



AppDynamics Apps

End-User Monitoring

Application Performance Monitoring

Infrastructure Visibility

App iQ Platform

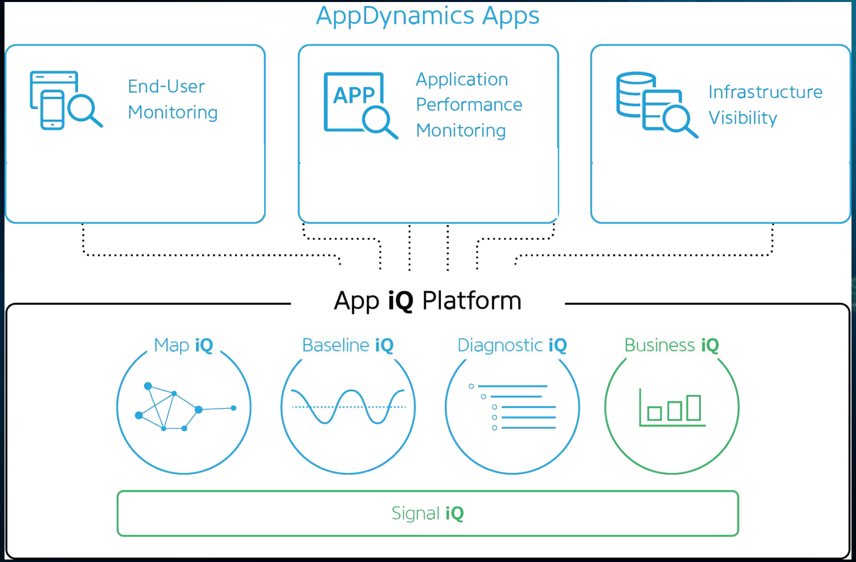
# App iQ Platform Summary

- Discover a business processing unit – BT
- Self learn its performance metrics – report constantly, all traffic
- Self learn its distributed activity – report constantly, all traffic

- Self learn its performance thresholds – evaluate constantly
- Self learn code level activity – report only when needed

- Automatic detection of SLA failures
- Self collect diagnostics – for anomalies and problem patterns

# App iQ is Extensible



Over 100 3rd Party Extensions

AppD Cloud

Private Cloud

# Key AppDynamics Differentiators

## Map iQ

End-to-end business transaction tracing

Full business context

## Baseline iQ

Machine learning

Alerting on deviations from baseline

## Diagnostic iQ

Deep diagnostic data to code-level

Low overhead, in high-scale, production environments

## Business iQ

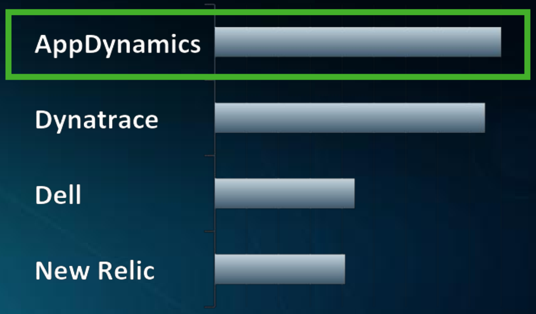
Real-time business monitoring and alerting

Correlate app performance and business impact

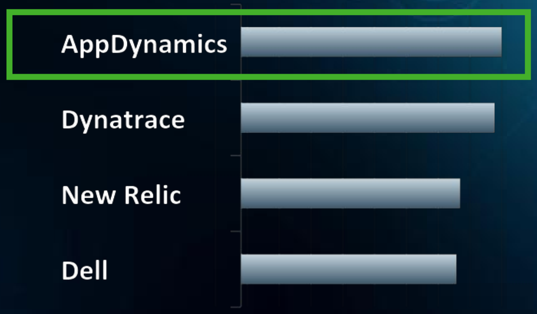
## Deployment Choice

# Gartner Rates AppDynamics #1 Across All Critical Capabilities

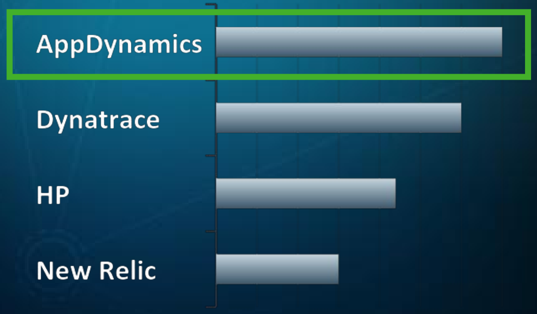
AppDynamics #1 – Application Support



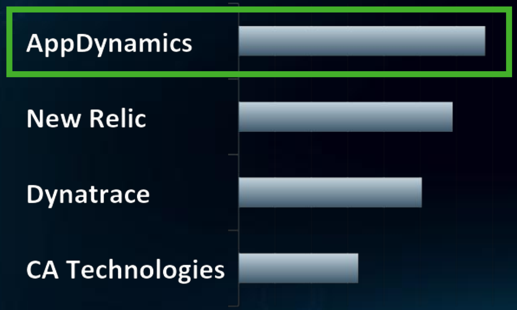
AppDynamics #1 – Application Development



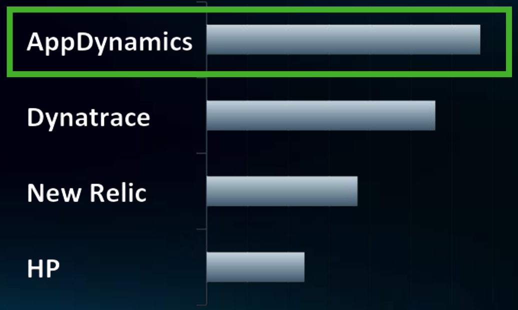
AppDynamics #1 – IT Operations



AppDynamics #1 – Line of Business



AppDynamics #1 – DevOps Release



*Source: Gartner Critical Capabilities for APM, 2016*

# The World's Leading Enterprises Use AppDynamics

| Financial Services | Industrial | Media and Entertainment | Tech/Telecom | Healthcare | Consumer | Government |
|---|---|---|---|---|---|---|
| UBS | Audi | comcast NBC UNIVERSAL | salesforce | BlueCross BlueShield | POLO RALPH LAUREN | UNITED STATES POSTAL SERVICE |
| Citizens Financial Group, Inc. | FedEx | Bloomberg | cisco | MAYO CLINIC | TESCO | UNITED STATES FEDERAL RESERVE SYSTEM |
| PayPal | United Airlines | DreamWorks | vodafone | McKESSON | Expedia | State of California Franchise Tax Board |
| Nasdaq | CHOICE HOTELS | sky MEDIA | eHarmony | KAISER PERMANENTE | WILLIAMS-SONOMA | CERN |
| Quicken Loans | SIEMENS | EA | okta | Pfizer | Shutterfly | UNITED STATES CUSTOMS AND BORDER PROTECTION |

| Partners |
|---|
| IBM |
| accenture |
| Capgemini |
| Cognizant |
| rackspace |

# “In their own words”

“Without AppDynamics, the transformation from being reactive to proactive would not have been possible!”
- The Container Store

“Business transactions gave the ING team an entirely new perspective on how to view requests across their distributed system.”
- ING

“AppDynamics provides historical data, so we can get automated baseline for normal performance and then trigger diagnostics if there’s a major deviation.”
- Priceline.com

“…it used to take several people looking at several different monitoring solutions, comparing numbers to find any problem with our system.”
- eHarmony

“AppDynamics is a game changer. It has transformed our applications into a living, breathing entity.”
- Pearson Education

“AppDynamics was rolled out globally across 23 data centers – deploying a total of 15,000 agents in just one week – all to a single AppDynamics report server.”
- Cisco

“It was simple to deploy. It took minutes, literally. In our proof of concept, we saw results within an hour.”
- NASDAQ

*Source: AppDynamics Website*

# Thank You