Caving and Big Data

By Damien Dallimore

Developer Evangelist

splunk>

About me

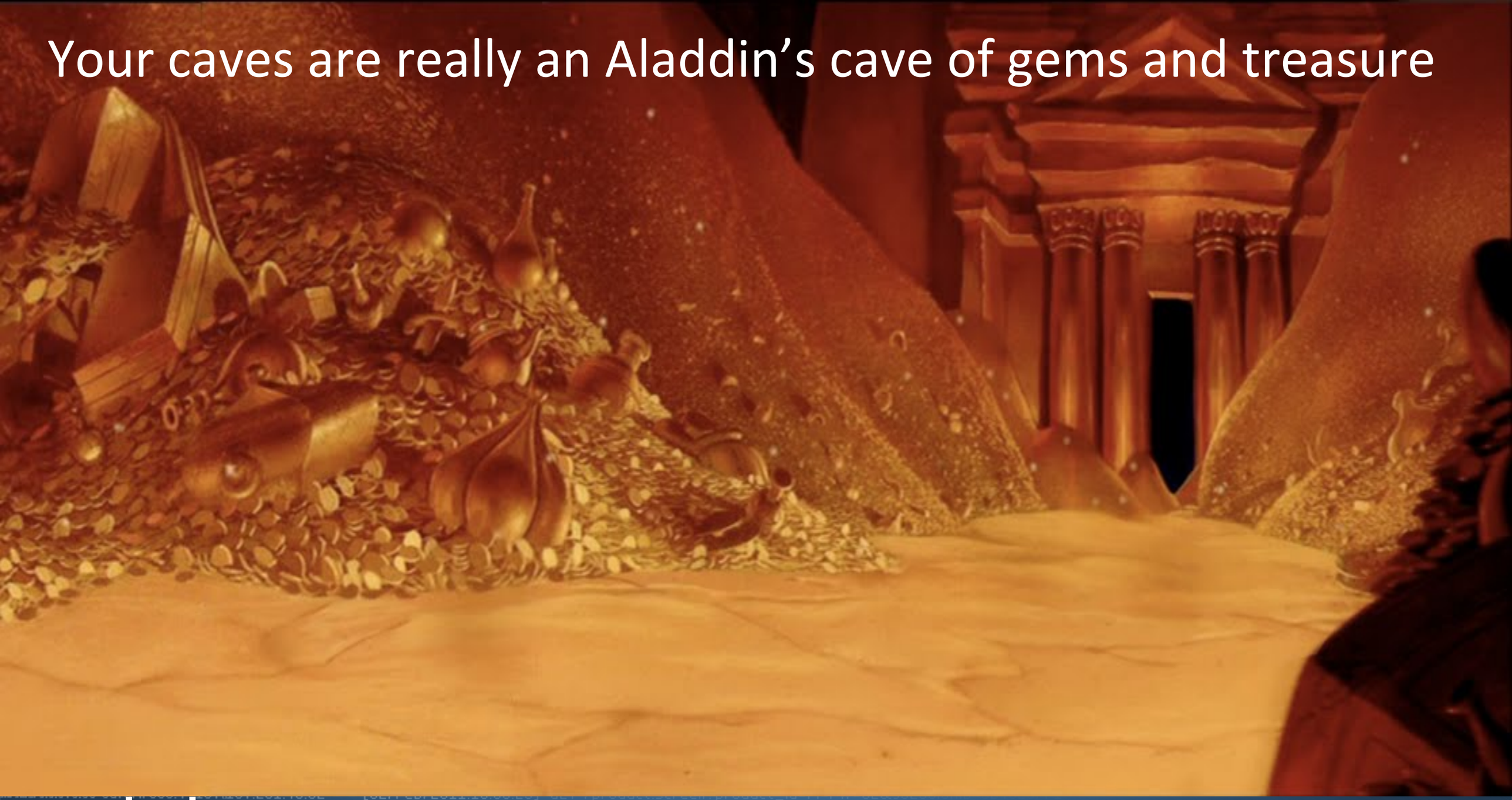
- Developer Evangelist at Splunk
  - Develop solutions and apps for the Splunk platform
  - Get out in the community and talk about it
- Splunk architect and community collaborator
  - I used to be a Splunk Customer
- Coder
  - Been paying my mortgage cutting code most of my career
- Kia Ora
  - I hail from Middle Earth aka New Zealand , now I tend to live out of a suitcase.

Lets go Spelunking !

But what does the Spelunking metaphor mean ?

- Massive amounts of “data” are being generated everyday
- We are often blind to where that data is and what that data is telling us , it lives buried away in a dark cave
- We can navigate our way through the murk and make discoveries if we have the right equipment and know how (big data tools and techniques)

Your caves are really an Aladdin's cave of gems and treasure

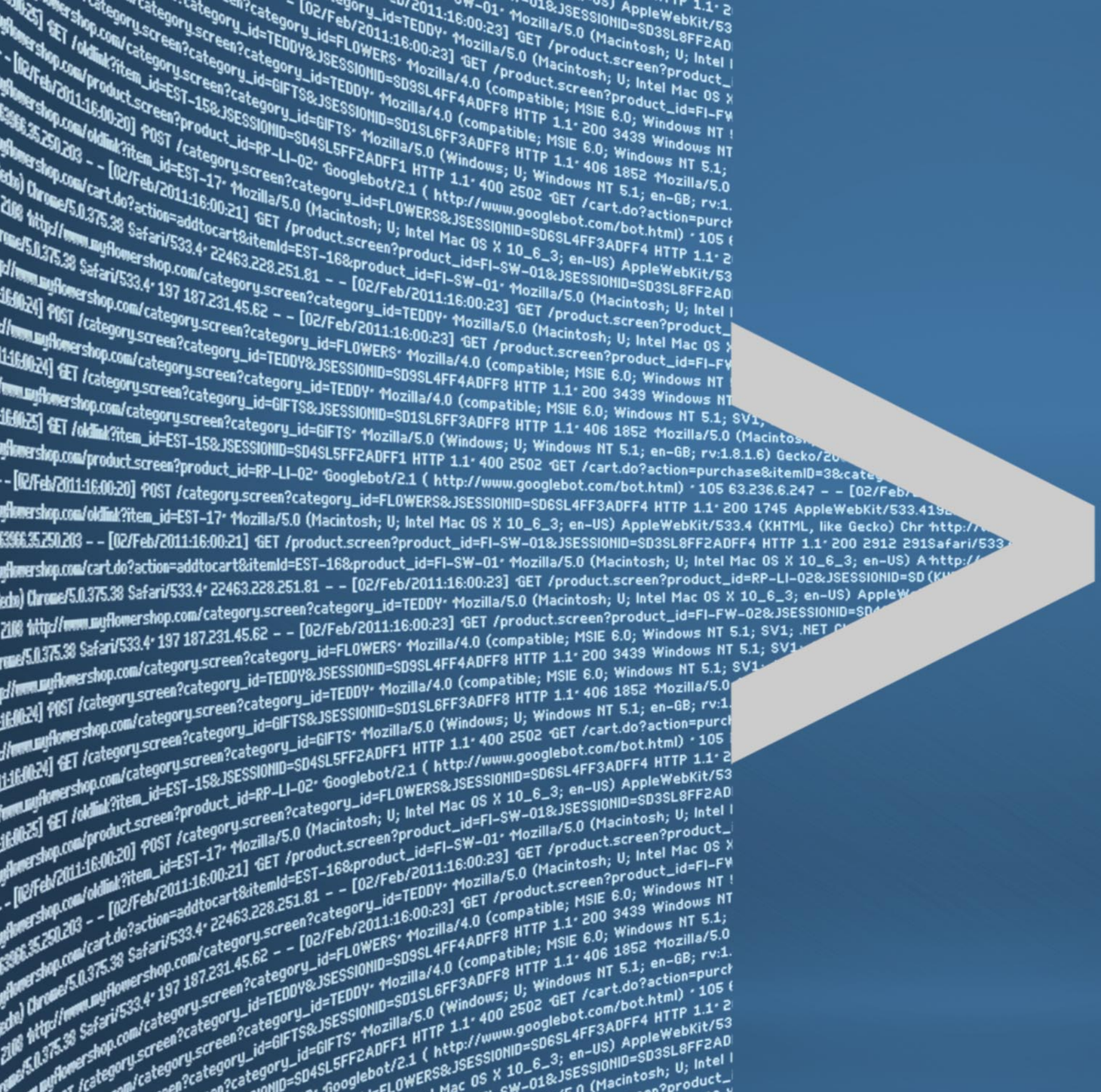


Agenda

- Overview of Splunk
- Splunk Developer platform
- Splunk Java SDK
- Spring Integration Splunk adaptors
- Monitoring JVM's
  - AppDynamics Integration overview

Along the way expect code and demos

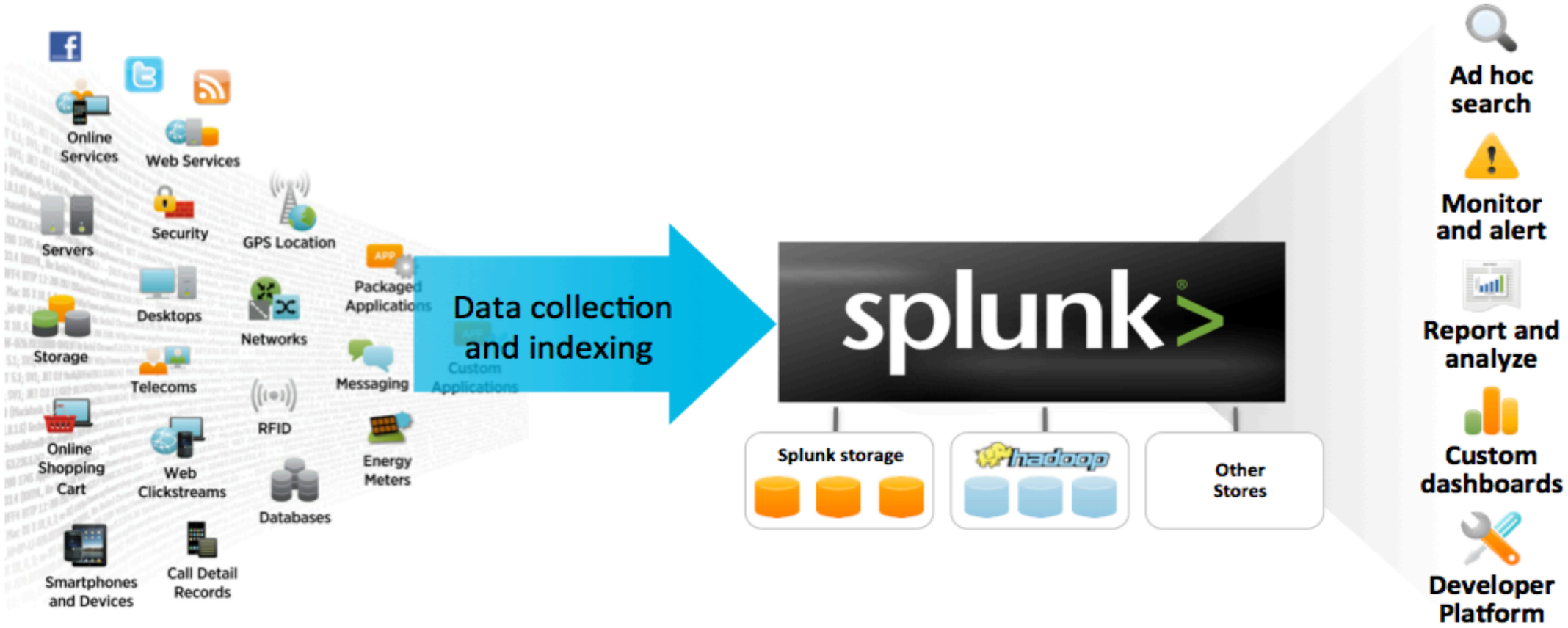
Questions , time depending, otherwise hunt me down afterwards



What is Splunk ?

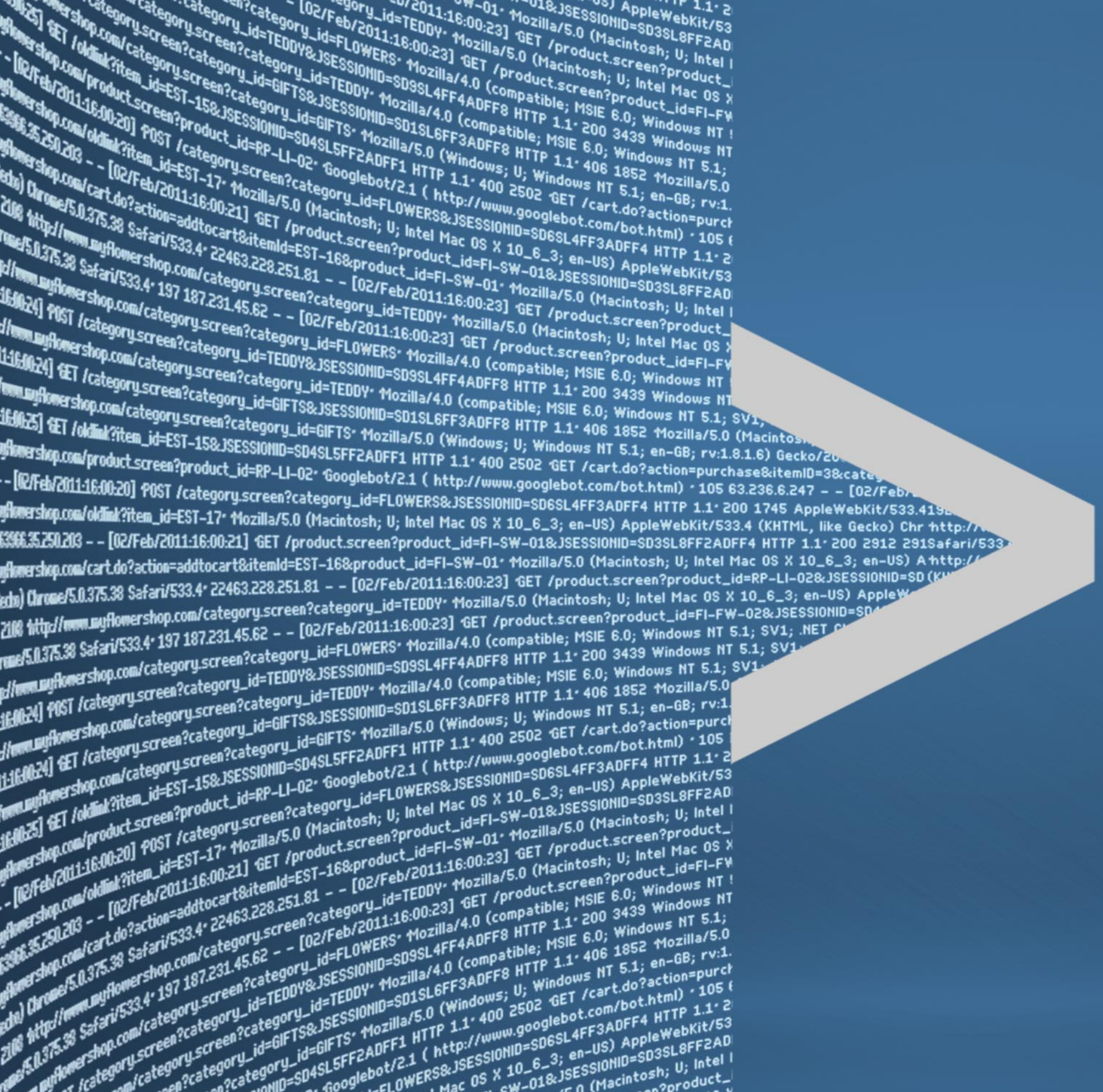
So What is Splunk, Exactly?

Splunk is a platform for all of your machine data



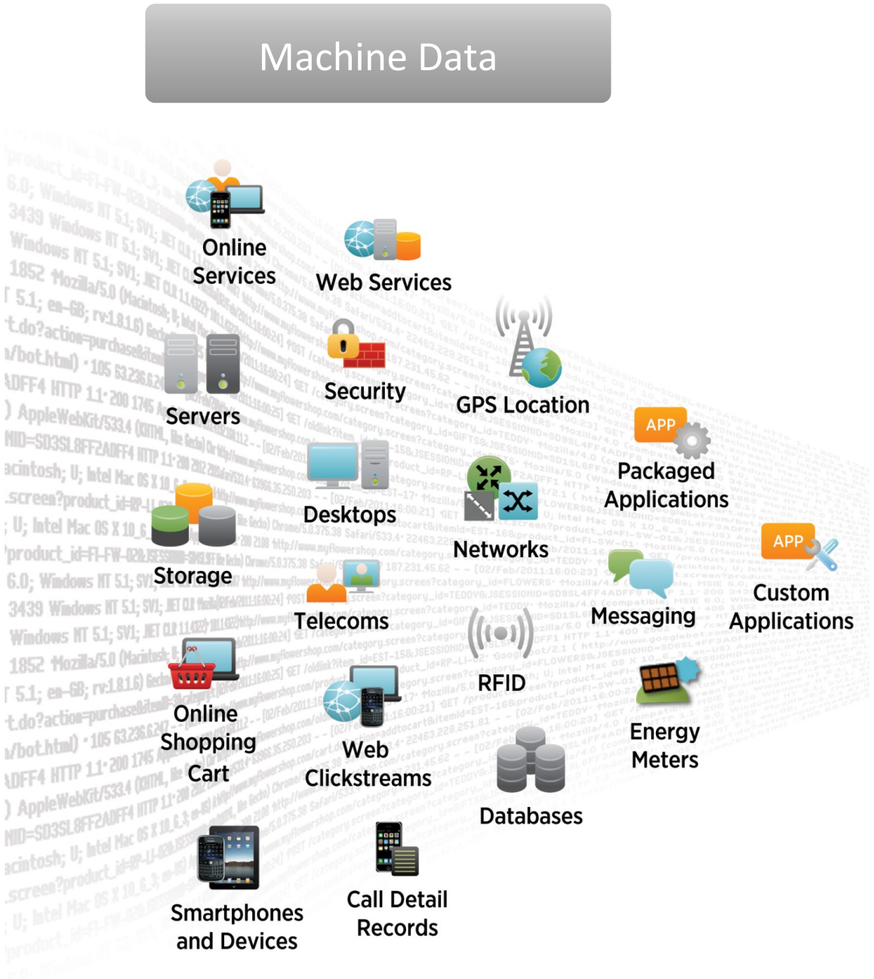
In a nutshell

- Integrated platform for data collection, searching & correlation and visualization
- Real time visibility of events, Real time alerting
- No need to write Map Reduce jobs
- Open and extensible
- Numerous ways to get your data into Splunk (TCP,UDP,REST,File monitoring, custom inputs etc..)
- “Schema on the fly”, no need to define/enforce structure up front
- Highly available distributed architecture allows Splunk to scale out to TB’s per day , Index replication
- Stream in data directly from your application code
- Provides comprehensive controls for data security, retention and integrity
- Splunkbase community , heaps of free apps and add-ons, community support
- Bi directional integration with other data sources (Hadoop, Casandra, Amazon S3 / Glacier, Mongo, RDBMS etc..)



Splunk Developer Platform

Splunk & Developers



SplunkUI (Splunk Apps)

Custom/ Existing Applications

SDKs

Search, chart and graph
Save and schedule searches as alerts
Export search results
Manage inputs and indexes
Add & remove users and roles

REST API

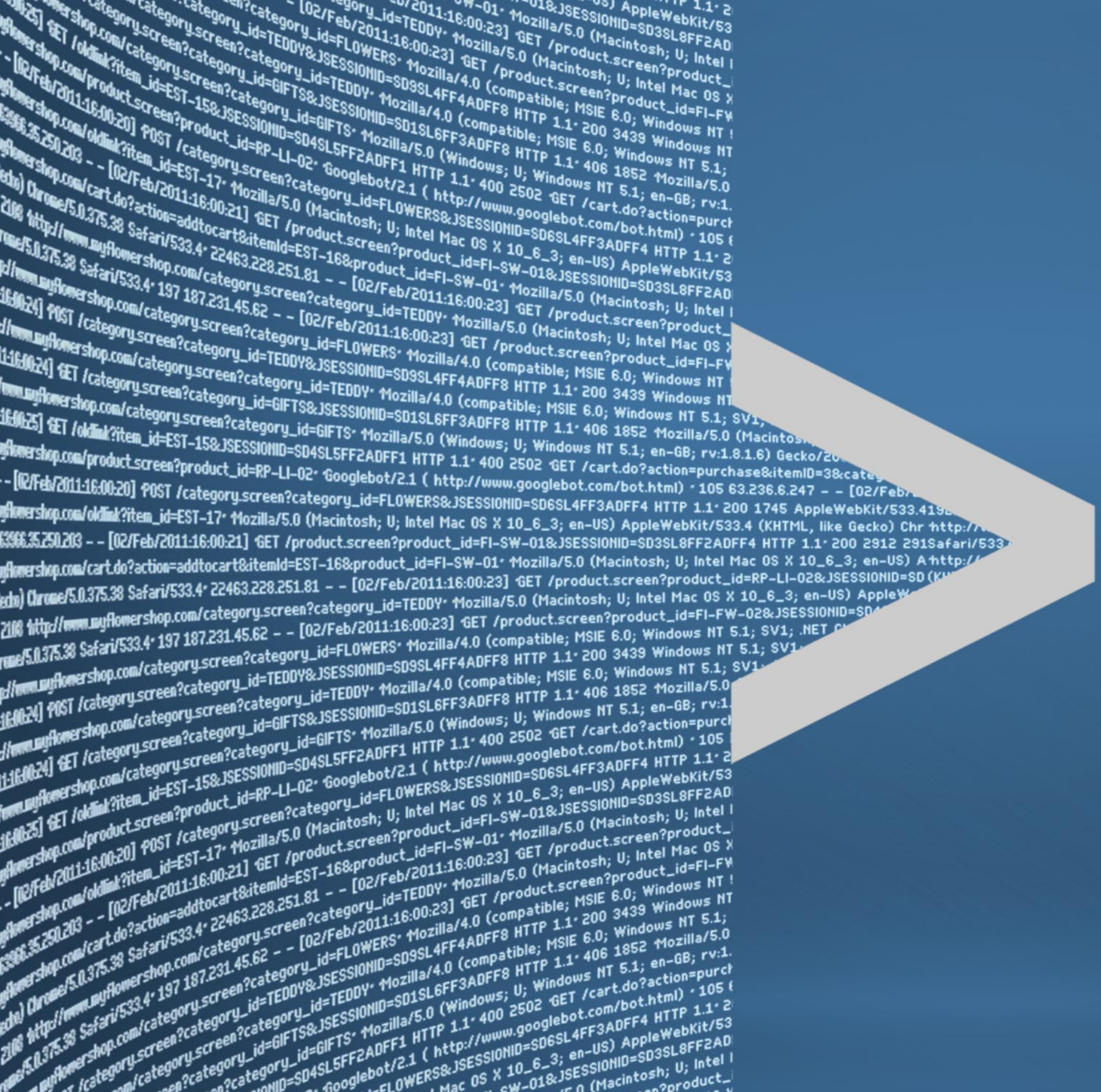
splunk> Engine

The Splunk REST API

- Exposes an API method for every feature in the product
  - Whatever you can do in the UI – you can do through the API
  - Run searches
  - Manage Splunk configurations
- API is RESTful
  - Endpoints are served by splunkd
  - Requests are GET, POST, and DELETE HTTP methods
  - Responses are Atom XML Feeds or JSON
  - Versioning Support
  - Search results can be output in CSV/JSON/XML/Raw
  - Authentication is token based

Developer Platform SDKs

- We want to make it as easy as possible for developers to build Big Data apps on top of the Splunk platform
- Several different language offerings, Software Development Kits (SDKs)
  - Javascript, Java, Python, PHP, Ruby(beta), C#(private repo)
- All Splunk functionality is accessible via our SDKs
- Get Data into Splunk
- Execute Splunk Searches, get data out of Splunk
- Manage Splunk
- Customized User Interfaces



Using Splunk for Application Development

Top 3 Developer Takeaways

- **Every** developer can use Splunk to accelerate dev & test and gain application intelligence
- The developer platform lets customers **customize** and **extend** the power of Splunk
- Splunk lets developers build big data apps with the **skills they already have**

Takeaway 1: Use Splunk to **accelerate** dev & test

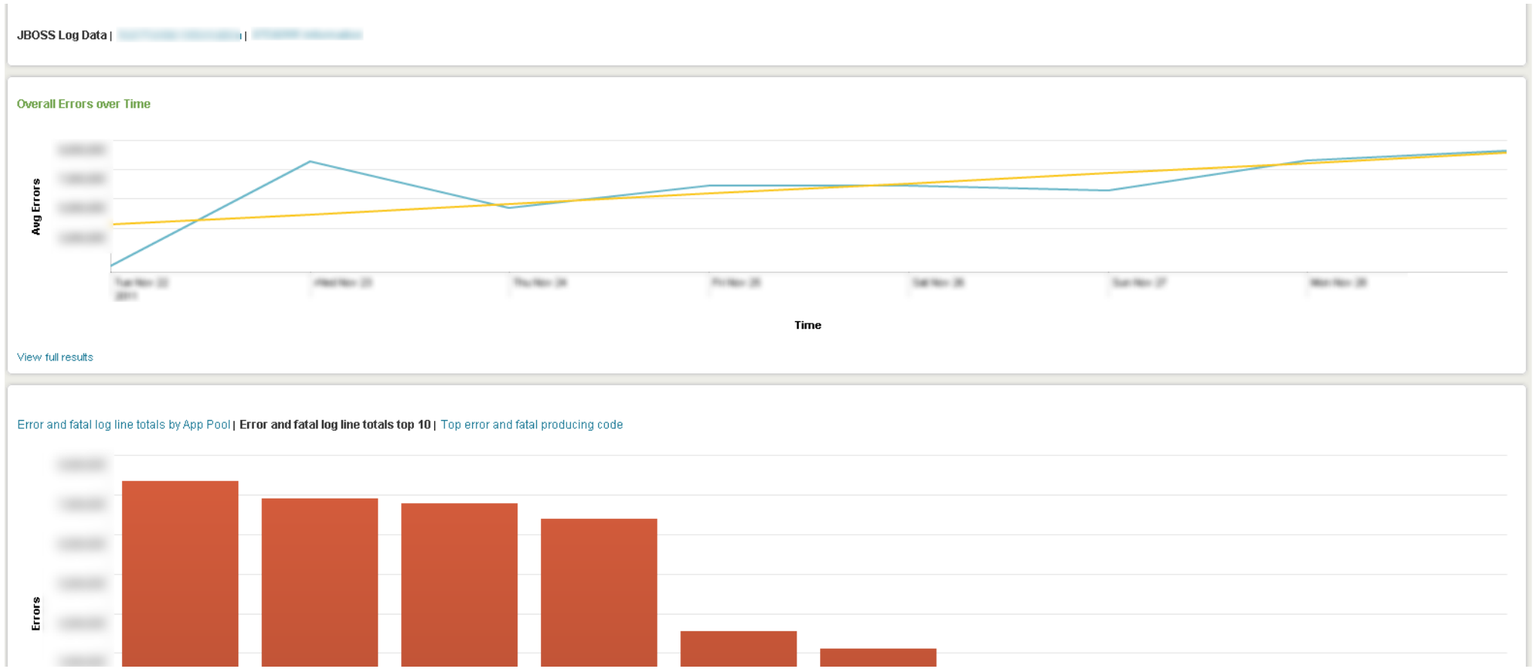
- Splunk frees you from upfront database design for analytics
  - late binding schema
- Developers and QA/test engineers don't have to ask IT/Ops to get logs off machines
  - Role base access to all data within one console without having to log into production systems
  - All events are indexed and accessible in real-time in one place.
  - Ad-Hoc real-time monitoring and historical investigation searchable from one place
  - Correlations and insights across multiple tiers.
- Splunk lets you find issues quickly, so you can fix issues quickly
- Integrate Splunk search results into testing assertions

Accelerate Dev & Test Cycles

"Splunk filled a vacuum we didn't know we had."

Nathan Pratt
Tech Lead, Tools & Automation

StubHub!



High-level view of application errors - used by site operations, engineering, and upper management

- Use case started with Site Operations to resolve issues
- Grew to include development and testing to improve application quality and time-to-release
- Release requirement – Projects are required to certify that all logs are Splunk-friendly (Semantic Logging)

Takeaway 2: **Customize** and **extend** Splunk

Integrate data from Splunk into existing apps and systems

Build custom line-of-business apps powered by Splunk

REST API & SDKs

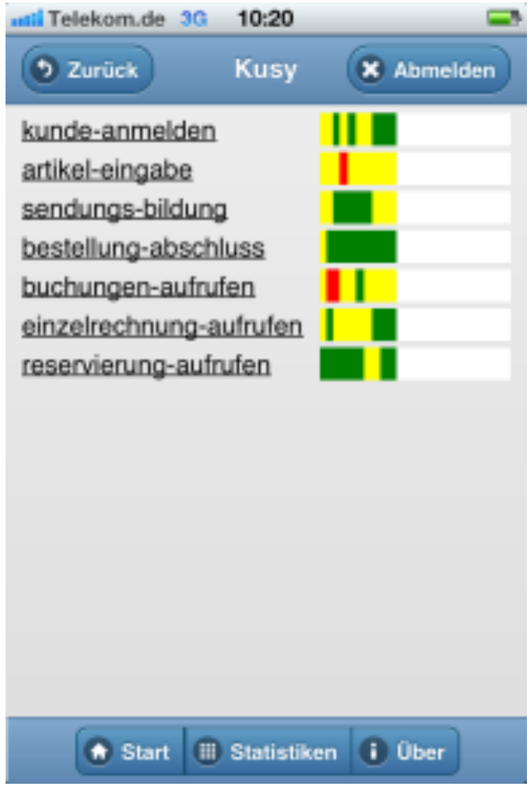
Deliver Operational Intelligence to marketing, sales, customer service and other divisions beyond IT in the systems and apps that make sense to them.

Integrating Splunk Data into a Mobile App

"The Splunk SDKs offer us so many ideas - we could use the JavaScript SDK in our own Operations Portal for showing Charts, Graphs, etc. "

Jens Ihnow
Applications Management
Back Office Systems

otto group



- Using the REST API to get key operational and business data out of Splunk
- One-touch views for Web Service Response Time and Top Sellers

Takeaway 3: Splunk lets developers build big data apps with the **skills they already have**

- Developers can use the languages and frameworks they know and love – like Python, JavaScript, Java and PHP.

- No need to write MapReduce jobs, learn R or be some kind of scientist to build apps that use Big Data – be a developer!

socialize

Splunks 7 million API calls per day and exposes Splunk data to customers in their customer-facing web app via REST API

Hurricane LABS

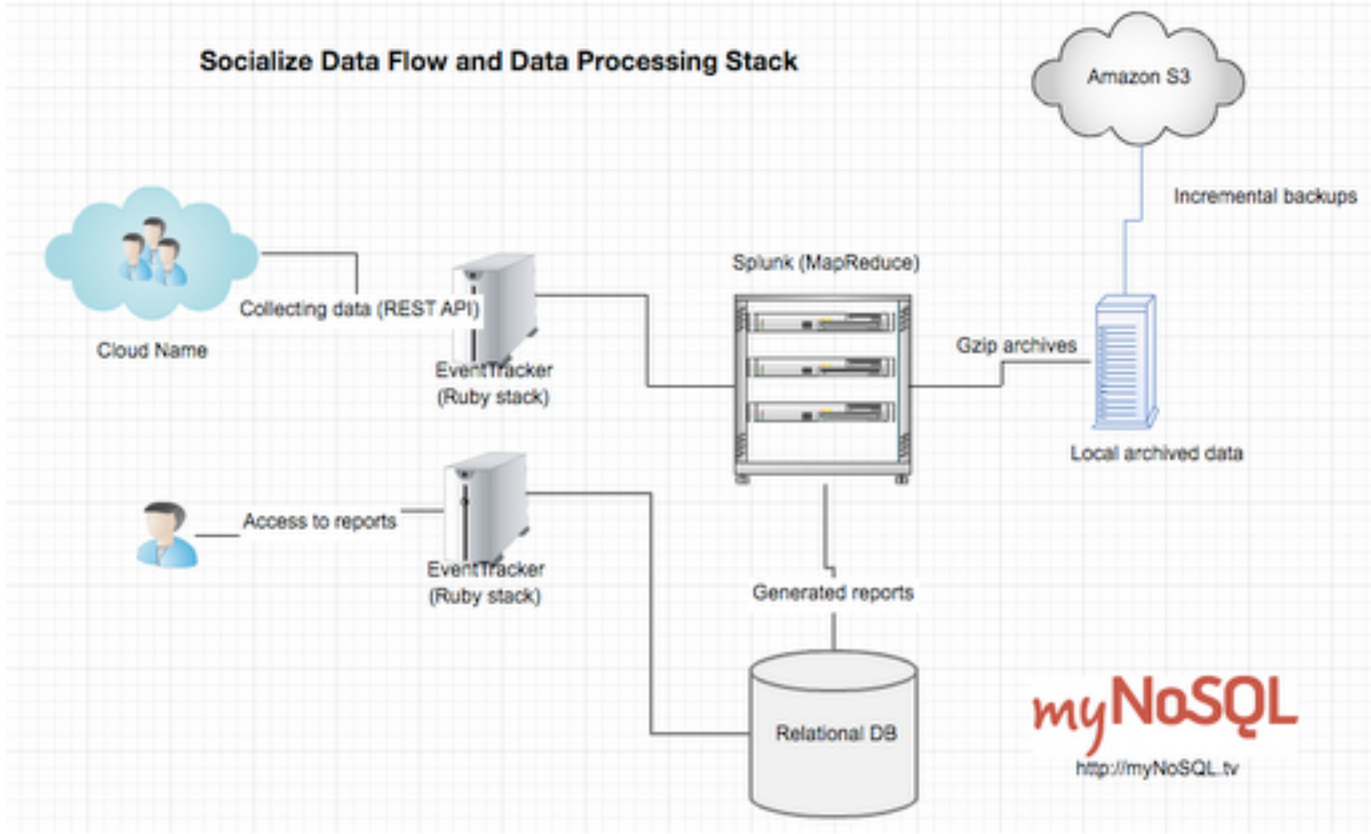
Using the Python SDK to deliver customers real-time security intelligence into custom dashboards

Building Big Data Apps on top of Splunk

"Splunk eliminates the need to write large MapReduce jobs to get meaningful information out of our data. This means we can get powerful stats and information to our key stakeholders in a fraction of the time."

Isaac Mosquera
CTO

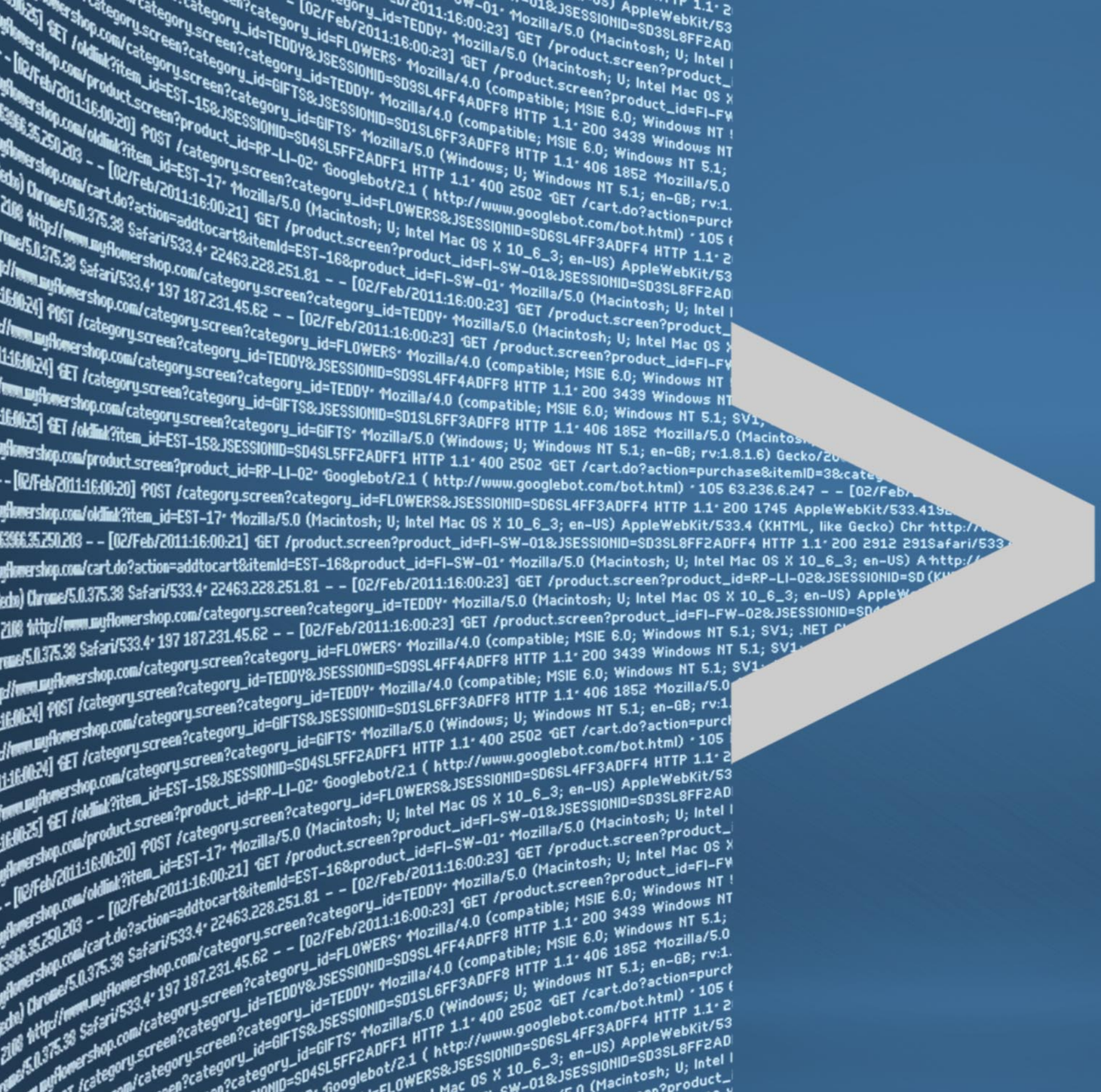
socialize



- Social marketing platform for mobile app developers, Splunking over 7 million API requests per day and 1 million actions per month
- Using the Splunk REST API to build custom dashboards for customers powered by data from Splunk

Why choose to develop on Splunk ?

- Splunk is not agnostic of its underlying data source , MapR algorithm optimized to Splunk index files
- Real time vs Batch Jobs
- Optimal for time series based data
- End to End Integrated Big Data Solution
- Fine grained protection of access and data using role based permissions
- Data retention and aging controls
- Users can submit “Map Reduce” jobs without needing to know how to code a MapR job
- Get the best of many worlds ie: Splunk Hadoop Connect
- Splunk integrates easily with other systems, developers can then just focus on developing against 1 single platform



Splunk Java SDK

Get the Java SDK

- Open sourced under the Apache v2.0 license
- Clone from Github : git clone https://github.com/splunk/splunk-sdk-java.git
- Project level support for Eclipse and Intellij IDE's
- Pre-requisites
  - JRE 6+
  - Ant
  - Splunk installed
- Loads of code examples
  - Project examples folder
  - Unit Tests
  - http://dev.splunk.com
  - http://gist.github.com/damiendallimore
- Comprehensive coverage of the REST API
- Tutorial videos available at http://dev.splunk.com

Connect and Authenticate

```
public static Service connectAndLoginToSplunkExample() {

  Map<String, Object> connectionArgs = new HashMap<String, Object>();
  connectionArgs.put("host", ”somehost");
  connectionArgs.put("username", ”admin");
  connectionArgs.put("password", ”foobar");
  connectionArgs.put("port", 8089);
  connectionArgs.put("scheme", "https");

  // will login and save the session key which gets put in the HTTP Authorization header
  Service splunkService = Service.connect(connectionArgs);
  return splunkService;


}
```

Search

```
public static void simpleSearchExample() {

  Service splunkService = connectAndLoginToSplunkExample();
  String searchQuery = "search error OR exception| head 10";
  Args queryArgs = new Args();
  queryArgs.put("earliest_time", "-3d@d");
  queryArgs.put("latest_time", "-1d@d");
  // perform the search , blocks here
  InputStream stream = splunkService.search(searchQuery, queryArgs);
  processInputStream(stream);

}
```

Realtime Search

```
public static void realTimeSearchExample() {

  Service splunkService = connectAndLoginToSplunkExample();

  Args queryArgs = new Args();

  queryArgs.put("earliest_time", "rt-5m");
  queryArgs.put("latest_time", "rt");

  // submit the job
  Job job = splunkService.getJobs().create("search index=spring exception OR error”, queryArgs);

…


}
```

Input Events

```
public static void logEventToSplunkExample() {

  Service splunkService = connectAndLoginToSplunkExample();
  // Get a Receiver object
  Receiver receiver = splunkService.getReceiver();

  // Set the sourcetype
  Args logArgs = new Args();
  logArgs.put("source", ”http-rest");
  logArgs.put("sourcetype", ”demo-example");

  // Log an event into the spring index
  receiver.log(”spring", logArgs, ”QCon London rocks");

}
```

Semantic Logging

Log anything that can add value when aggregated, charted or further analyzed

Example Bogus Pseudo-Code:

```
void submitPurchase(purchaseId)
{
        log.info("action=submitPurchaseStart, purchaseId=%d", purchaseId)
        //these calls throw an exception on error
        submitToCreditCard(...)
        generateInvoice(...)
        generateFullfillmentOrder(...)
        log.info("action=submitPurchaseCompleted, purchaseId=%d", purchaseId)
 }
```

- Create Human Readable Events
- Clearly Timestamp Events
- Use Key-Value Pairs (JSON Logging)
- Separate Multi-Value Events
- Log Unique Identifiers

SplunkJavaLogging

- A logging framework to allow developers to as seamlessly as possible integrate Splunk best practice logging semantics into their code and transport events directly to Splunk.
- Custom handler/appender implementations(REST and Raw TCP) for the 3 most prevalent Java logging frameworks in play. Splunk events directly from your code.
  - LogBack
  - Log4j
  - java.util.logging

- Uses the Splunk Java SDK under the hood
- Better handling of stacktraces
- All code and examples is on Github

Developers just log as they are used to

```java
/**
 * Just log as usual, and wire up a Splunk REST/TCP appender to forward the
 * event to Splunk
 */
private static void simpleLogExample() {

    // get your logger
    Logger logger = LoggerFactory.getLogger("splunk.logger");

    // log a regular string
    logger.info("REST for the wicked");

}
```

Better

```java
/**
 * Format the log message to adhere to Splunk best practice logging
 * semantics
 */
private static void splunkLogEventExample() {

    // get your logger
    Logger logger = LoggerFactory.getLogger("splunk.logger");

    // create a SplunkLogEvent with a date and values quoted
    SplunkLogEvent event = new SplunkLogEvent("Failed Login", "someID");

    //add Splunk CIM fields
    event.setAuthApp("myapp");
    event.setAuthUser("jane");

    // add a custom field
    event.addPair("somefieldname", "foobar");

    // log a splunk log event generated string
    logger.info(event.toString());

}
```

A-HA

Search

```
index=main sourcetype=logback name="Failed Login" | stats count as "Failed Logins" by app,user
```

1 result in the last 15 minutes (from 3:46:00 PM to 4:01:59 PM on Tuesday, August 7, 2012)

Export Options

Overlay: None

	app	user	Failed Logins
1	myapp	jane	3

```
2012-08-07 15:54:06:644+1200 name="Failed Login" event_id="someID" app="myapp" user="jane" somefieldname="foobar"
```

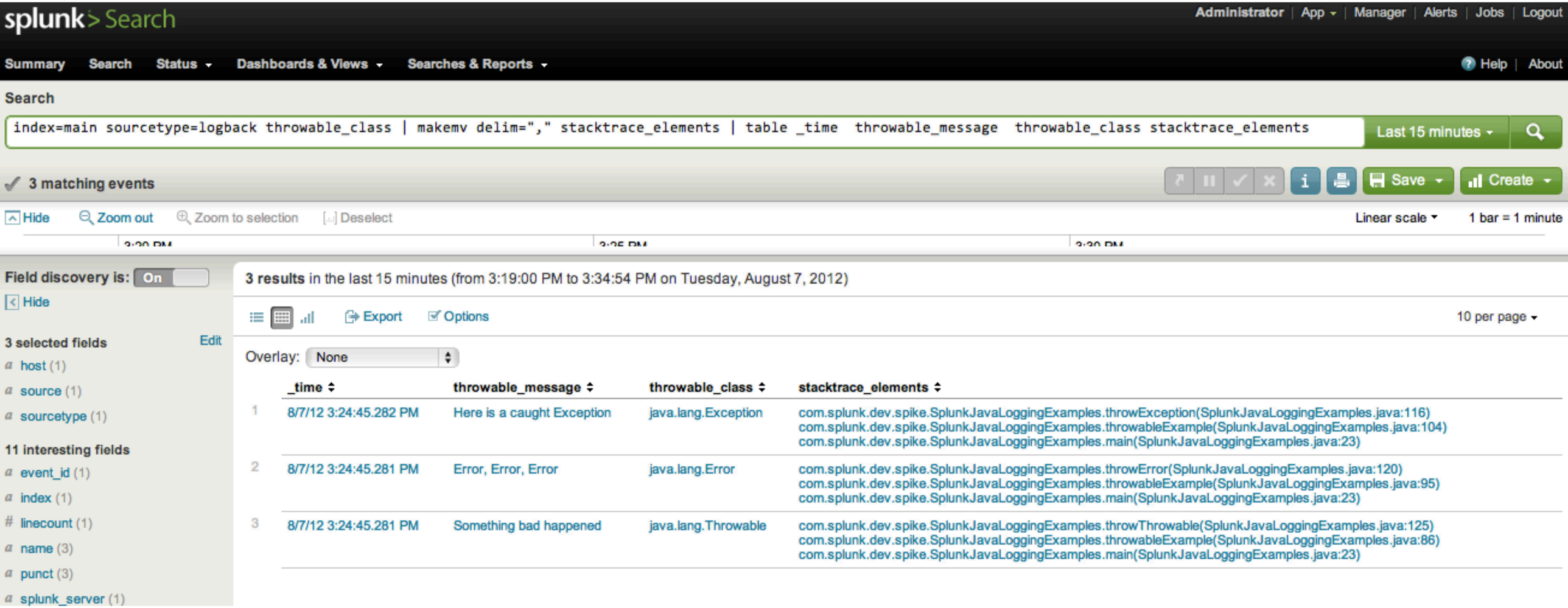
<barf>Typical Java Stacktraces in logs</barf>

```
java.lang.Throwable: Something bad happened
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.throwThrowable(SplunkJavaLoggingExamples.java:126)
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.throwableExample(SplunkJavaLoggingExamples.java:86)
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.main(SplunkJavaLoggingExamples.java:23)
java.lang.Error: Error, Error, Error
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.throwError(SplunkJavaLoggingExamples.java:121)
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.throwableExample(SplunkJavaLoggingExamples.java:96)
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.main(SplunkJavaLoggingExamples.java:23)
java.lang.Exception: Here is a caught Exception
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.throwException(SplunkJavaLoggingExamples.java:117)
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.throwableExample(SplunkJavaLoggingExamples.java:105)
        at com.splunk.dev.spike.SplunkJavaLoggingExamples.main(SplunkJavaLoggingExamples.java:23)
```

SplunkJavaLogging is your friend

```
/**
 * Log an Error/Exception/Throwable and handle the stacktrace elements in Splunk as a multi value field
 */
private static void throwableExample() {

        // get your logger
        Logger logger = LoggerFactory.getLogger("splunk.logger");

        try {

                throw new Exception("Danger Danger");

        } catch (Throwable throwable) {
                SplunkLogEvent event = new SplunkLogEvent();
                event.addThrowable(throwable);
                logger.error(event.toString());
        }

}
```

Java Stacktraces in Splunk



	_time	throwable_message	throwable_class	stacktrace_elements
1	8/7/12 3:24:45.282 PM	Here is a caught Exception	java.lang.Exception	com.splunk.dev.spike.SplunkJavaLoggingExamples.throwException(SplunkJavaLoggingExamples.java:116) com.splunk.dev.spike.SplunkJavaLoggingExamples.throwableExample(SplunkJavaLoggingExamples.java:104) com.splunk.dev.spike.SplunkJavaLoggingExamples.main(SplunkJavaLoggingExamples.java:23)
2	8/7/12 3:24:45.281 PM	Error, Error, Error	java.lang.Error	com.splunk.dev.spike.SplunkJavaLoggingExamples.throwError(SplunkJavaLoggingExamples.java:120) com.splunk.dev.spike.SplunkJavaLoggingExamples.throwableExample(SplunkJavaLoggingExamples.java:95) com.splunk.dev.spike.SplunkJavaLoggingExamples.main(SplunkJavaLoggingExamples.java:23)
3	8/7/12 3:24:45.281 PM	Something bad happened	java.lang.Throwable	com.splunk.dev.spike.SplunkJavaLoggingExamples.throwThrowable(SplunkJavaLoggingExamples.java:125) com.splunk.dev.spike.SplunkJavaLoggingExamples.throwableExample(SplunkJavaLoggingExamples.java:86) com.splunk.dev.spike.SplunkJavaLoggingExamples.main(SplunkJavaLoggingExamples.java:23)

Alternate JVM Languages

Scala	Groovy	Clojure
Javascript(Rhino)	JRuby	PHP(Quercus)
Ceylon	Kotlin	Jython

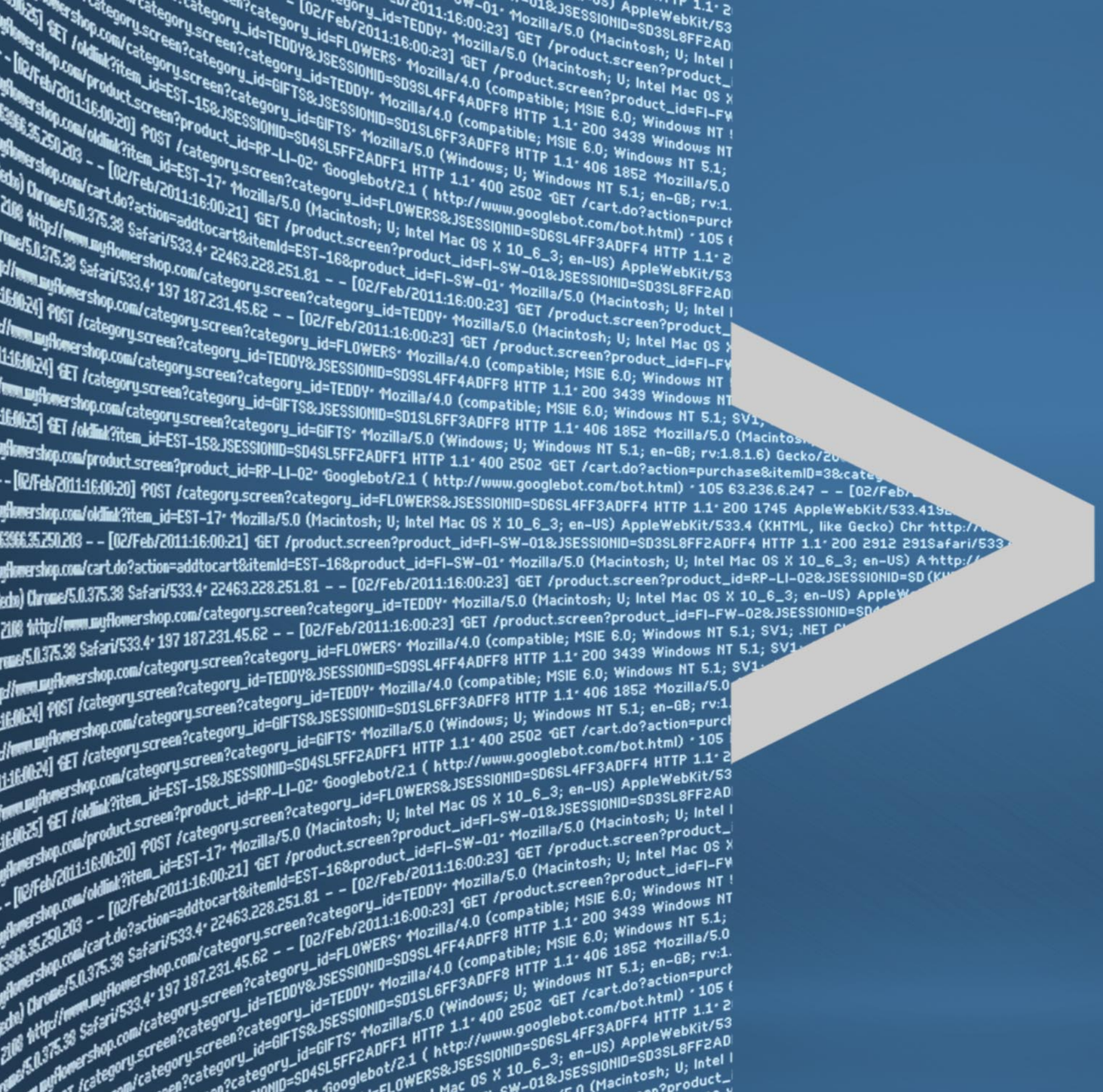
We don’t need SDK’s for these languages , we can just use the Java SDK !

Groovy

```
class SplunkJavaSDKWrapper {

  static main(args) {
    //connect and login
    def connectionParameters = [host:”somehost",username:”admin",password:”somepass"]
    Service service = Service.connect(connectionParameters)
    //get Splunk Server info
    ServiceInfo info = service.getInfo()

    def splunkInfo = [:]

    for (key in info.keySet())
      splunkInfo.put(key,info.get(key))

    printSplunkInfo(splunkInfo)

  }
  static printSplunkInfo(splunkInfo) {
    println "Info”
    splunkInfo.each { key, value ->println key + " : " + value}
  }
}
```

Scala

```
import com.splunk.Service._
import scala.collection.mutable.HashMap
import scala.collection.JavaConversions._

object SplunkJavaSDKWrapper {

  def main(args: Array[String]) = {
    //connect and login
    val connectionArgs = HashMap[String, Object]("host" ->"somehost","username" ->"me","password" ->"foo")
    val service = connect(connectionArgs)
    //get Splunk Server info
    val info = service.getInfo
    // Scala/Java conversion
    val javaSet = info.keySet
    val scalaSet = javaSet.toSet
    //print out Splunk Server info
    for (key <- scalaSet)
      println(key + ":" + info.get(key))
  }
}
```



Spring Integration
Splunk Adaptors

Spring Integration

- Spring Integration is an extension to core Spring
  - Based on *"Enterprise Integration Patterns"* model
  - Messaging model and Declarative Adaptors
  - Makes it easier to build integration solutions

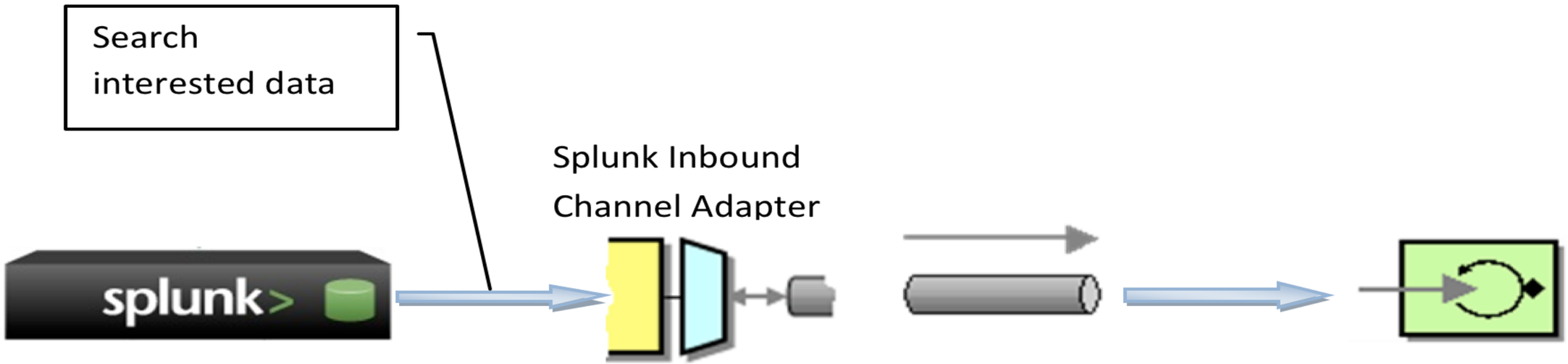
Spring Integration Splunk Adaptors

- Splunk Java SDK makes it easier to use the REST API
- Building on this , the Spring Integration Adaptors make it easier for Spring/Java developers to declaratively build data integration solutions and utilize the power of the Splunk platform

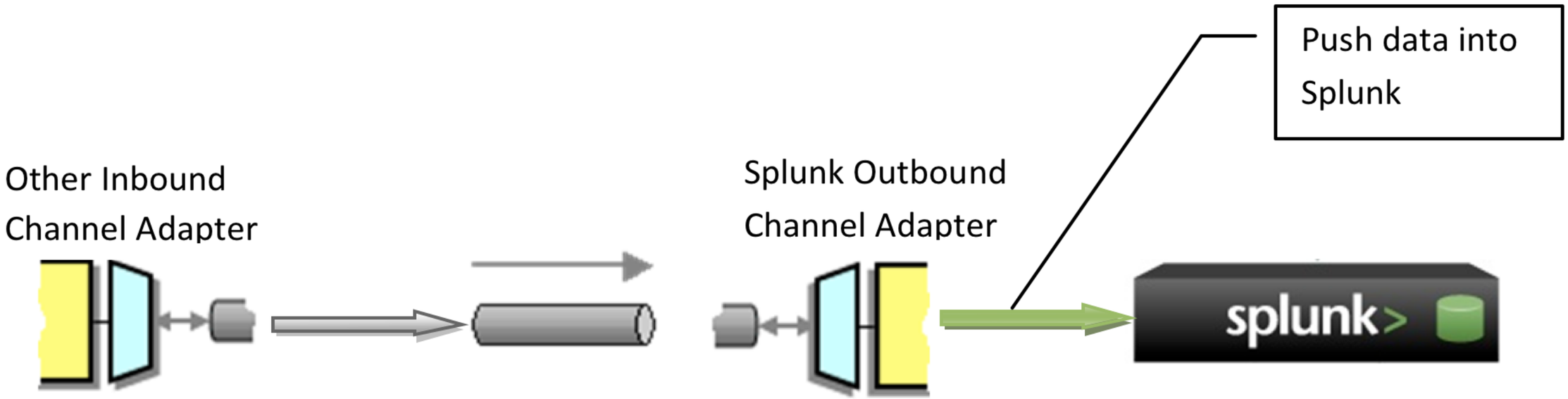
- https://github.com/SpringSource/spring-integration-extensions

- **Inbound Adaptor**
  - Search and export the data from Splunk and push into message channels
  - Filter, transform, export to other destinations
- **Outbound Adaptor**
  - Can consume data acquired by other Integration adaptors(Twitter, JDBC…) and push it into Splunk for indexing, searching and visualization

Spring Integration Splunk Inbound Adaptor



Spring Integration Splunk Outbound Adaptor



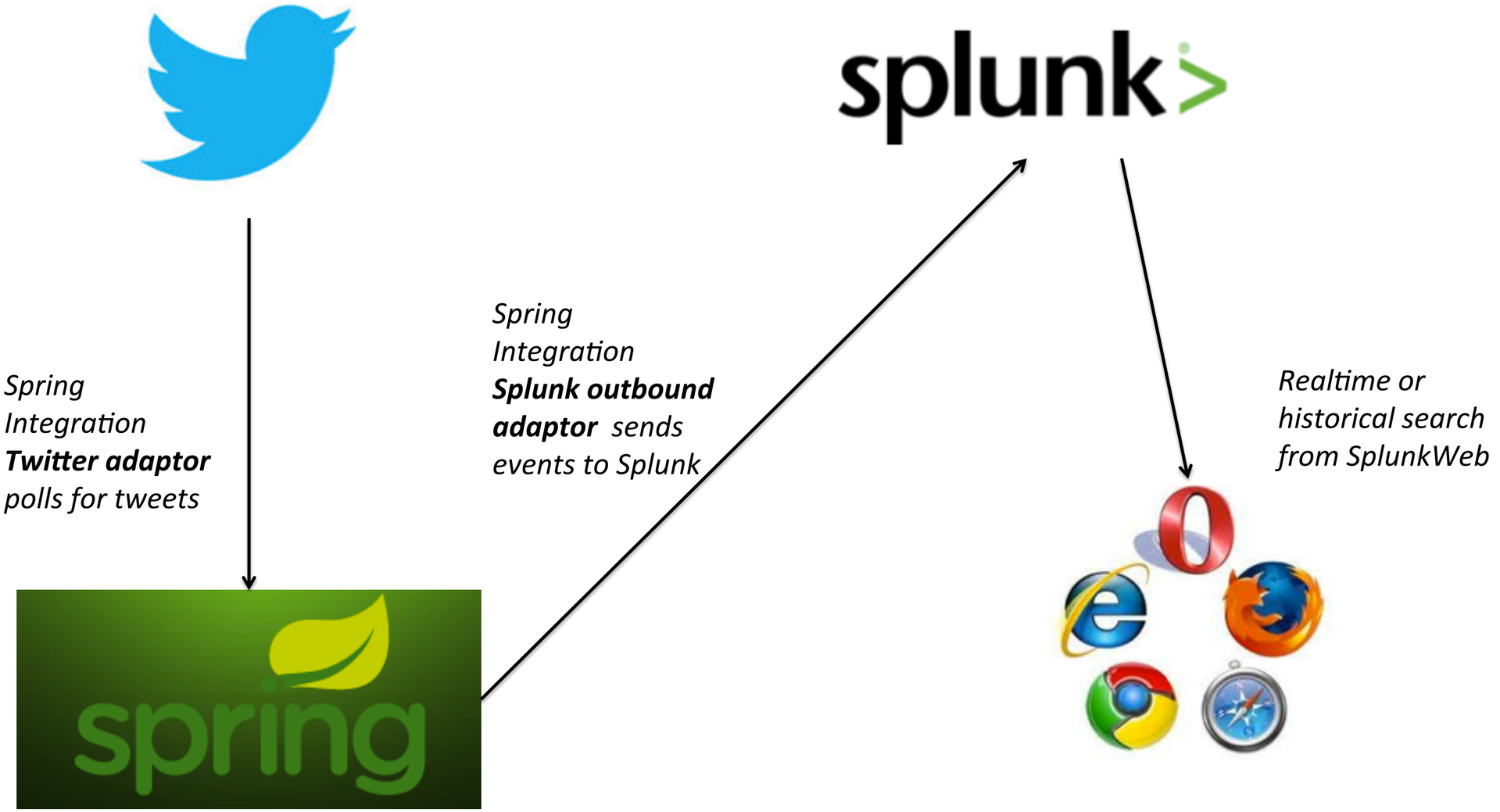
- HTTP REST Input
- TCP Input

Realtime Developer Integration Demo

WARNING , AUDIENCE PARTICIPATION AHEAD

Tweet some messages with the hashtags **#qconlondon** or **#splunk**

- Custom Java program is polling Twitter looking for the above tag
- This Java program is using the Spring Splunk Integration Adaptors
- These adaptors use the Splunk Java SDK internally to search from and write to Splunk programmatically
- Program receives the tweets, transforms them and then writes them to Splunk
- We can then create some **realtime** visualizations in SplunkWeb

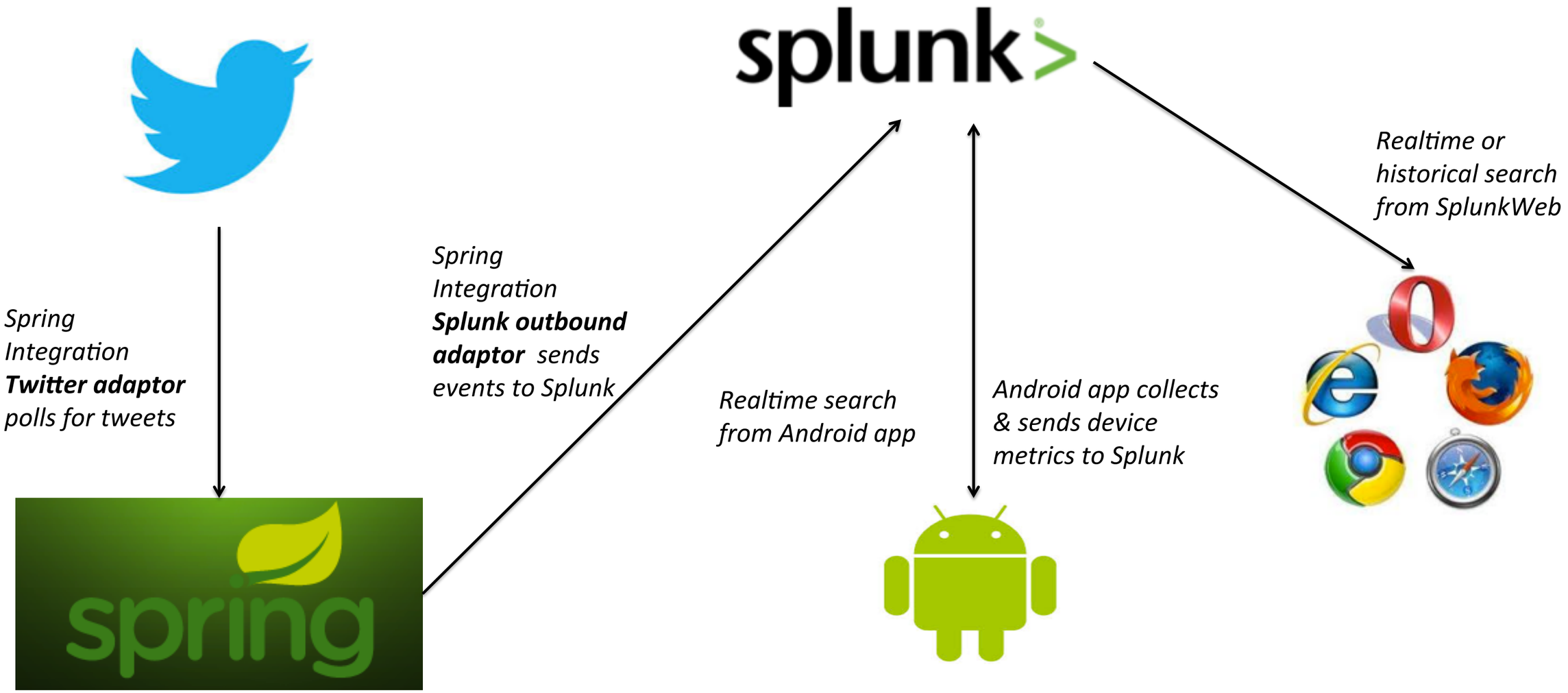


Raw events from Twitter are transformed into best practice logging format

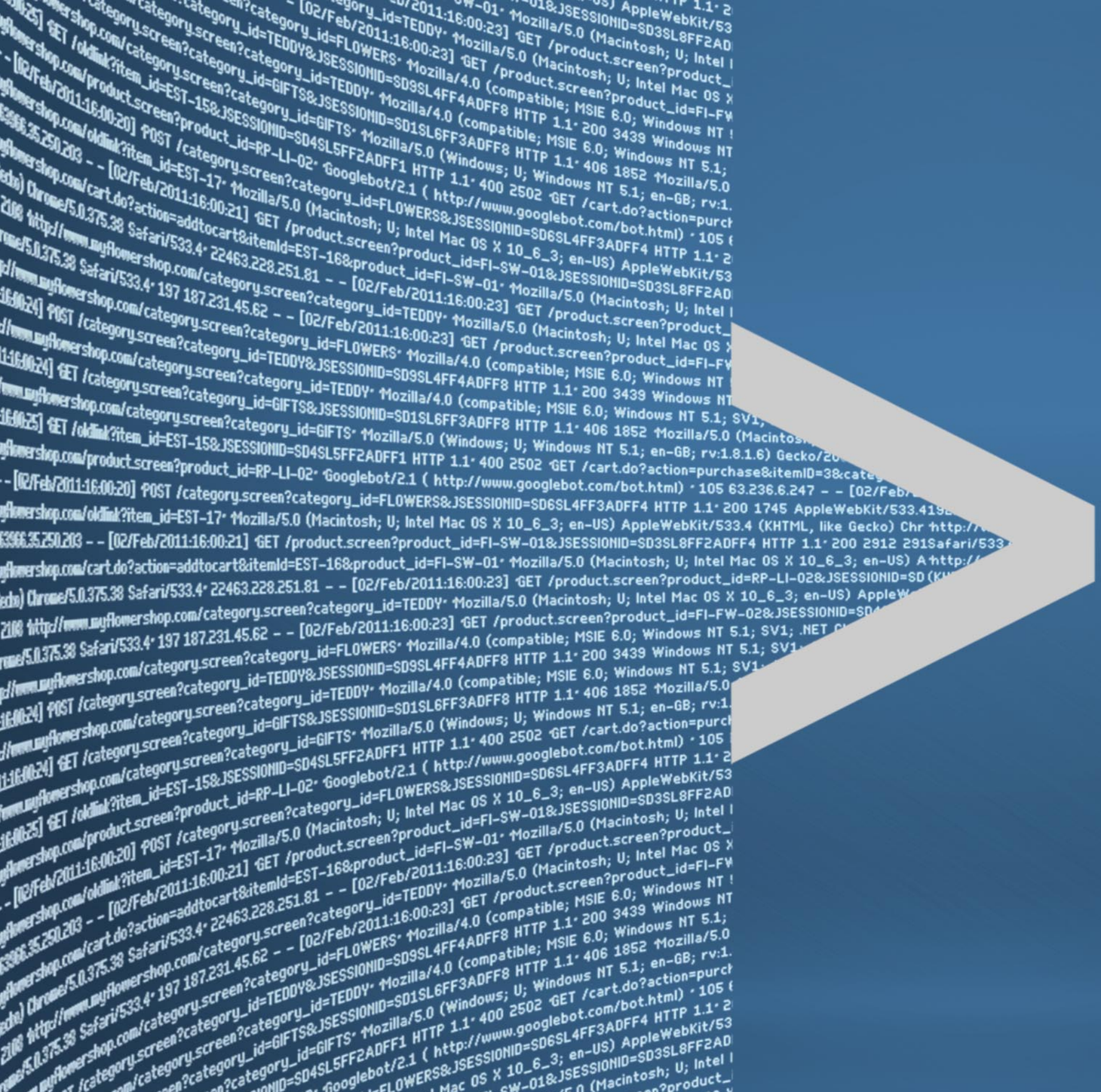
But I have a cooler idea

Android App Demo

- Custom developed Android app
- Uses the Splunk Java SDK , recompiled for Dalvik (Android's Java runtime) , expect a proper github branch soon.
- Streams realtime Splunk search results to device and displays some simple output
- Background service is pulling metrics of the device and sending to Splunk
- 800 million active Android devices by 2013 = big mobile data , big mobile developer community



Raw events from Twitter are transformed into best practice logging format



JVM Monitoring

JVM based systems today

JVM = Java Virtual Machine

Application Servers	Enterprise Service Buses	Databases
NoSQL	Distributed Big Data	Web Servers
Directory Servers	Search Engines	Build Systems
Gaming Platforms	Trading Systems	Reservation Systems
Core Banking	Messaging Infrastructure	Proprietary Systems

What is running in JVMs ?

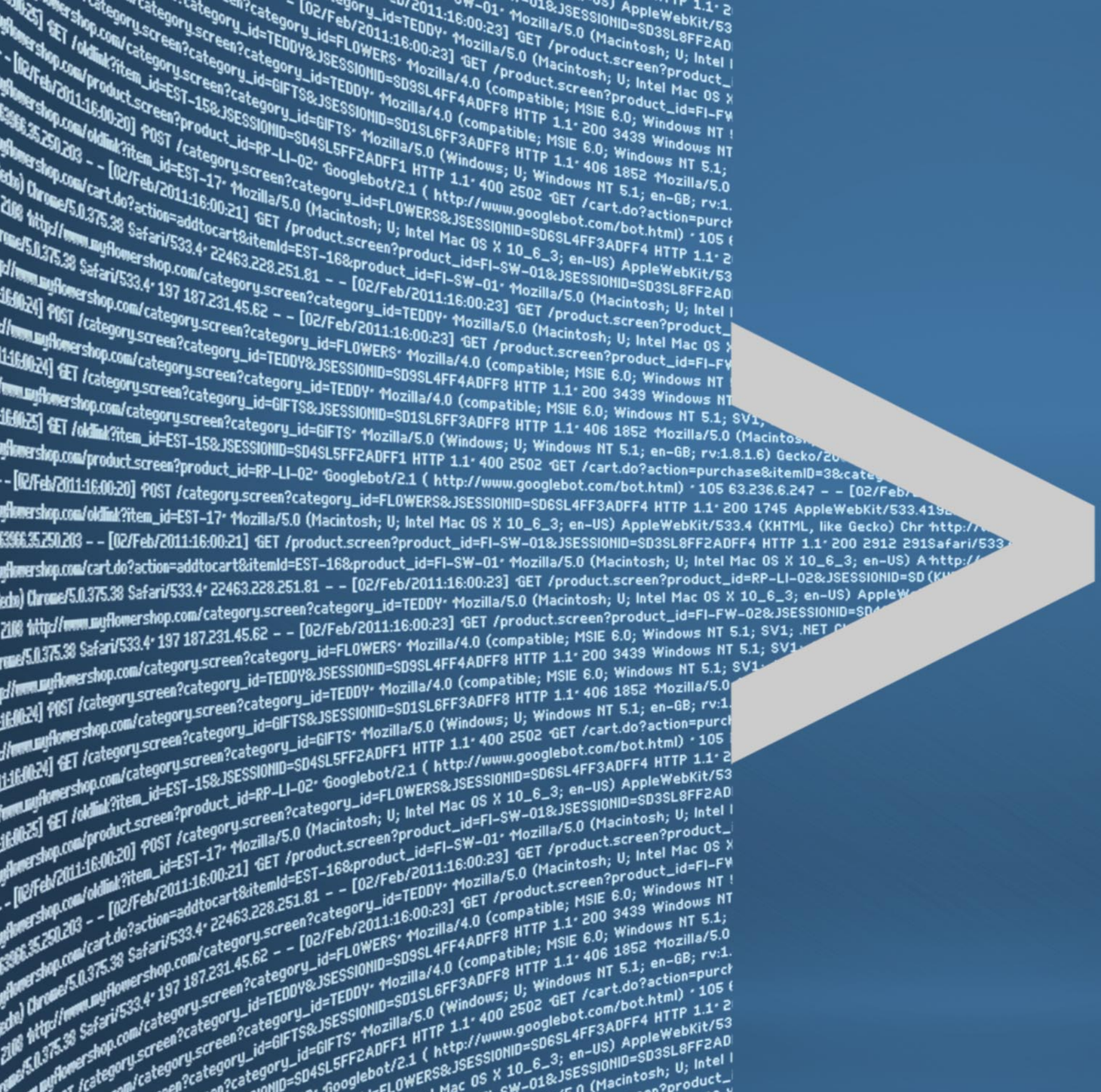
How can we monitor JVMs

- JVM log files (garbage collection) and binary dumps (hprof)
- Operating System metrics
- Application log files
- Using the Java SDK or SplunkJavaLogging to send events to Splunk
- Custom Agents (Byte Code Instrumentation)
- SNMP (if you really have to)

- **JMX (Java Management Extensions)**
  - In my experience , many of the useful operational metrics will come via JMX

Splunk for JMX

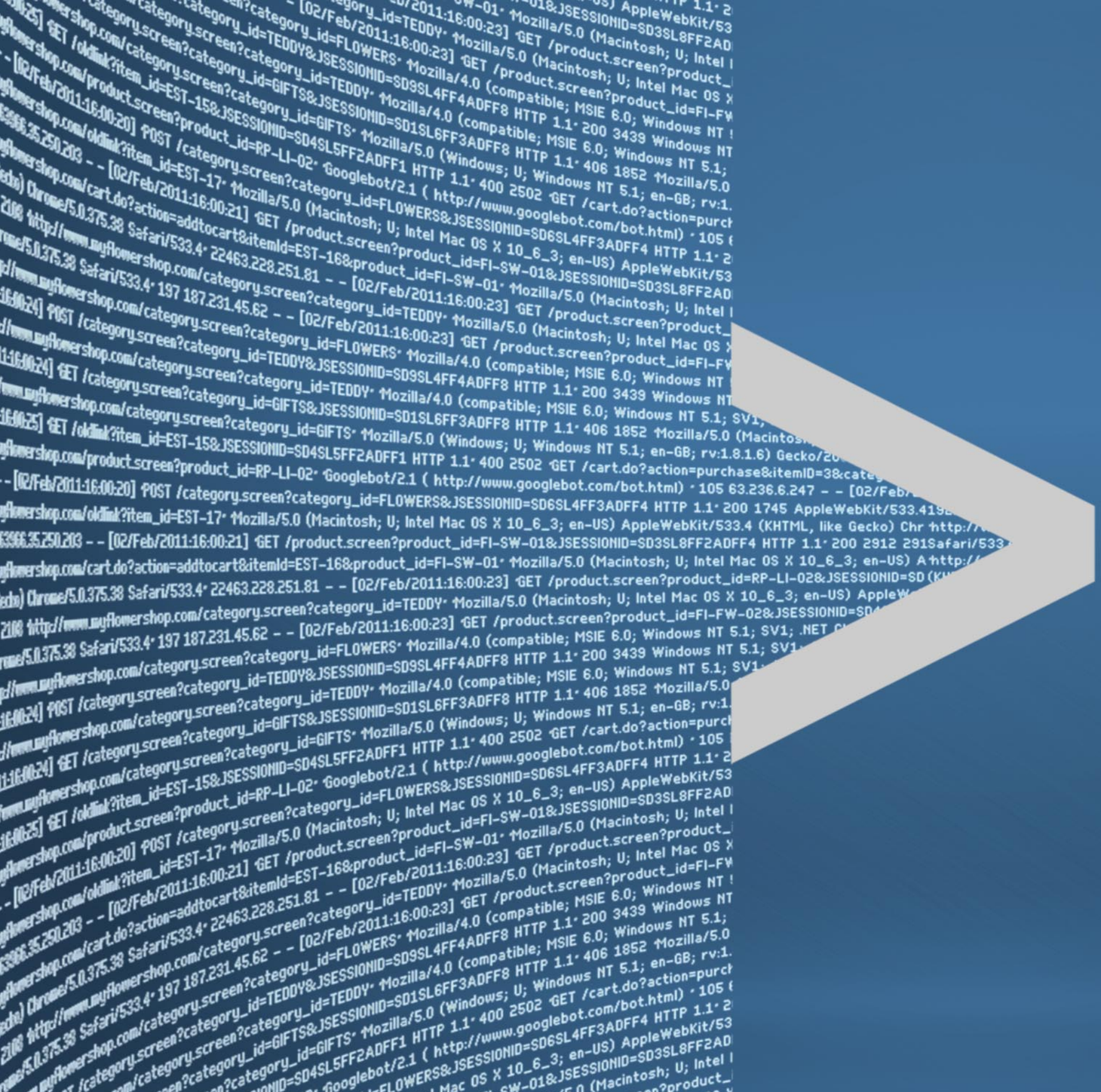
- SplunkBase App for monitoring JVM Applications
- Out of the box dashboards for JVM level monitoring (java.lang domain)
  - Memory , Threading, GC, CPU etc…
- Very simple configuration to wire up monitoring of any Mbeans from applications (Tomcat, JBoss, Cassandra, Coherence etc…)
- Hotspot, JRockit, IBMJ9, OpenJDK
- Poll JMX attributes and operations , index data over time, correlate with other data
- Supports large scale deployments of JVMs
- Extensible and Customizable
- Many connectivity options
  - RMI , IIOP, Direct Process Attachment, MX4J Hessian, Burlap and Soap
- Freely available download from SplunkBase & all code is on Github



Splunk for JMX Demo

JMS Messaging Input

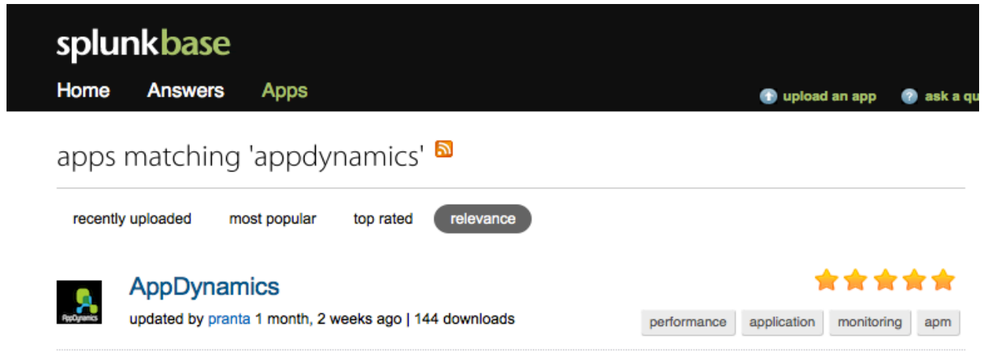
- JMS is simply a messaging interface that abstracts your underlying MOM provider implementation
- Most MOM vendors support JMS
- So this allowed for creating 1 single modular input that can index messages from :
  - MQ Series / Websphere MQ
  - Tibco EMS
  - ActiveMQ
  - HornetQ
  - RabbitMQ
  - SonicMQ
  - JBoss Messaging
  - Weblogic JMS
  - Native JMS
  - StormMQ
  - MSMQ (with a bit of stuffing around)
  - Etc…
- Simple to install : download from Splunkbase, drop in your apps directory, restart Splunk

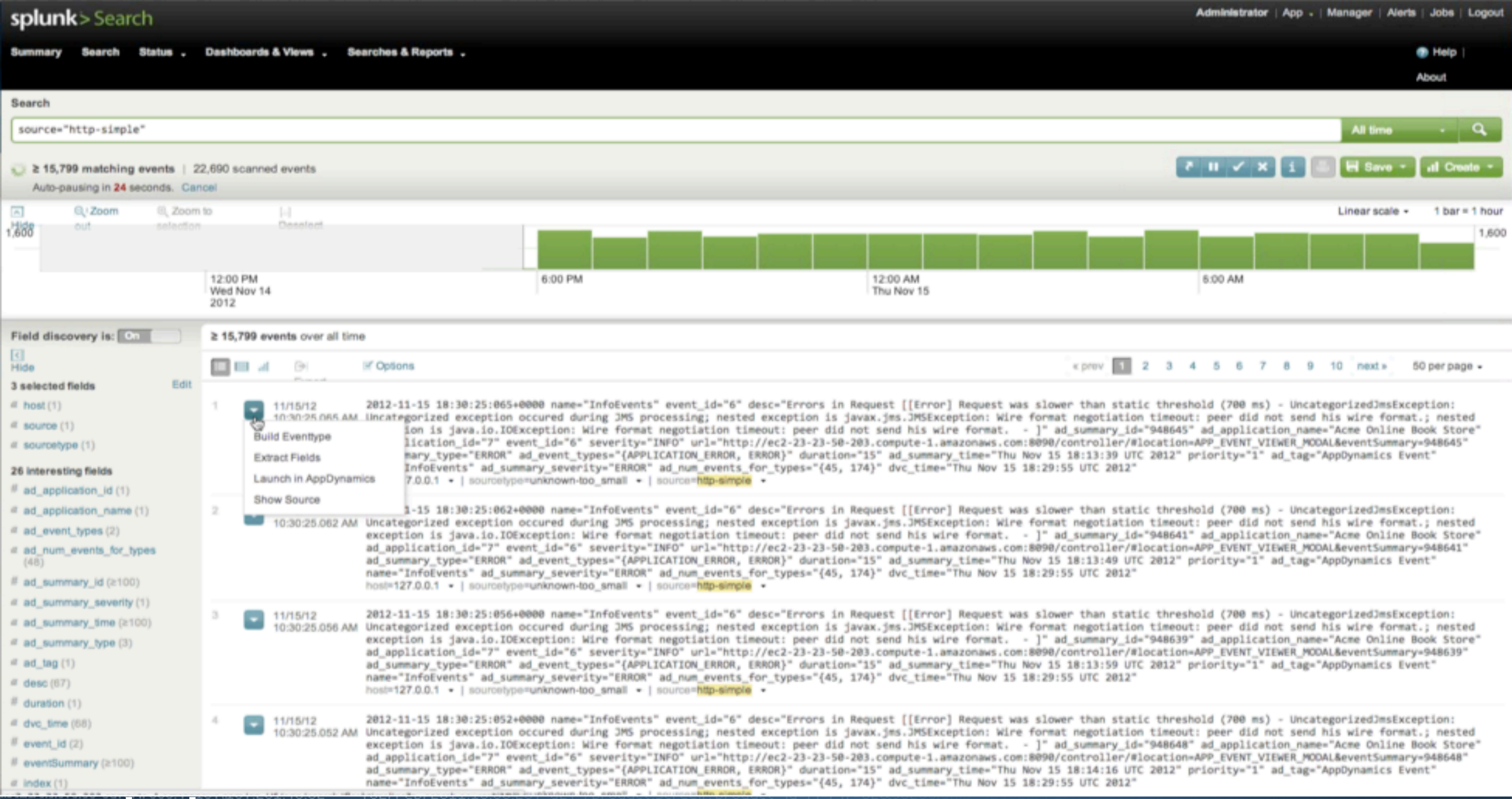


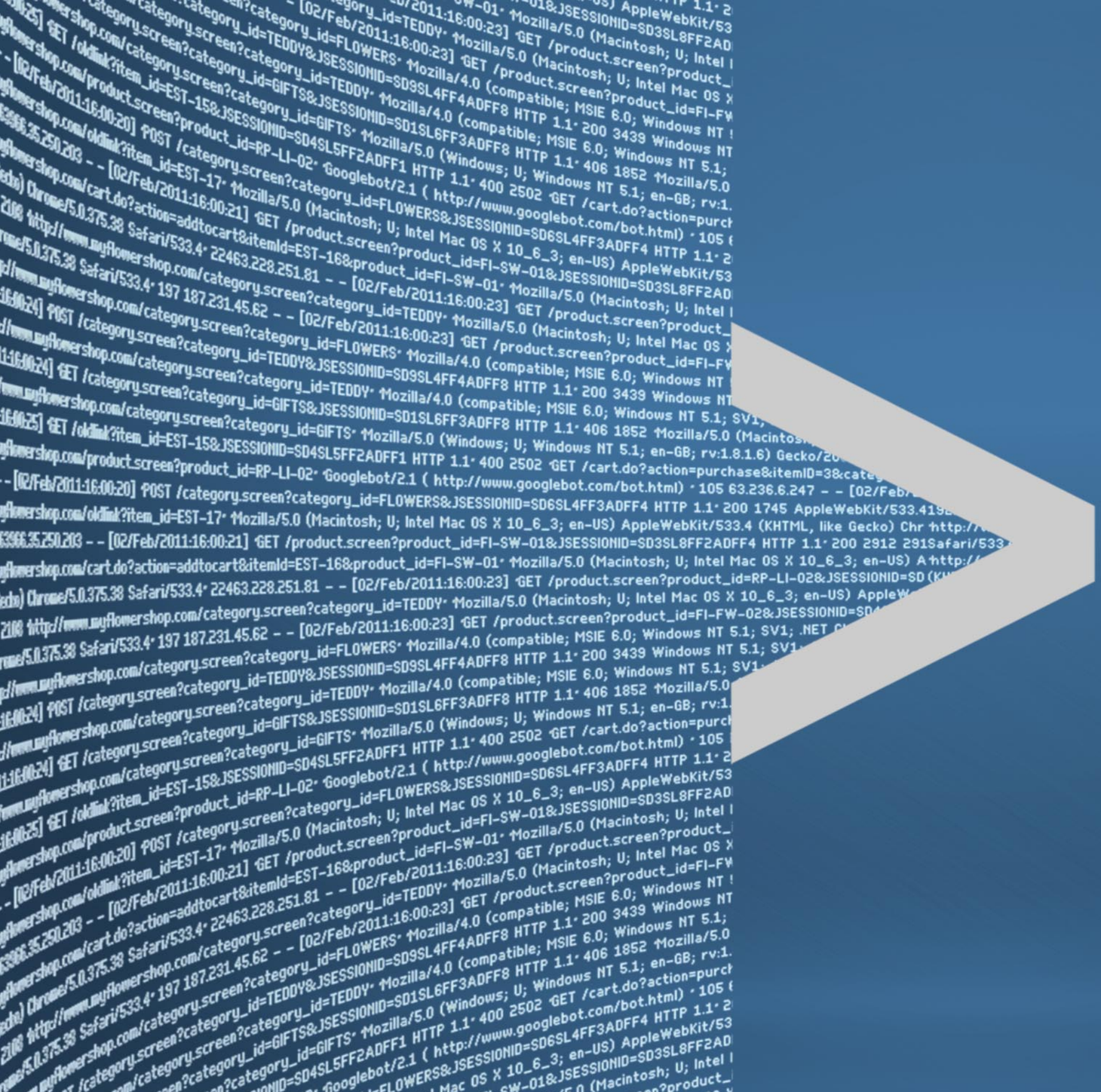
JMS Input Demo

AppDynamics integration

- Splunk is a platform that can integrate with other systems and sources of data
- Splunk indexes application performance monitoring data from AppDynamics using its REST API.
- AppDynamics pushes notifications on policy violations and events to Splunk
- Free Splunkbase app
- Correlate Appdynamics events with other data sources in Splunk
-
- Expose AppDynamics application performance metrics in Splunk dashboards and reporting
- Cross launch back into Appdynamics from Splunk

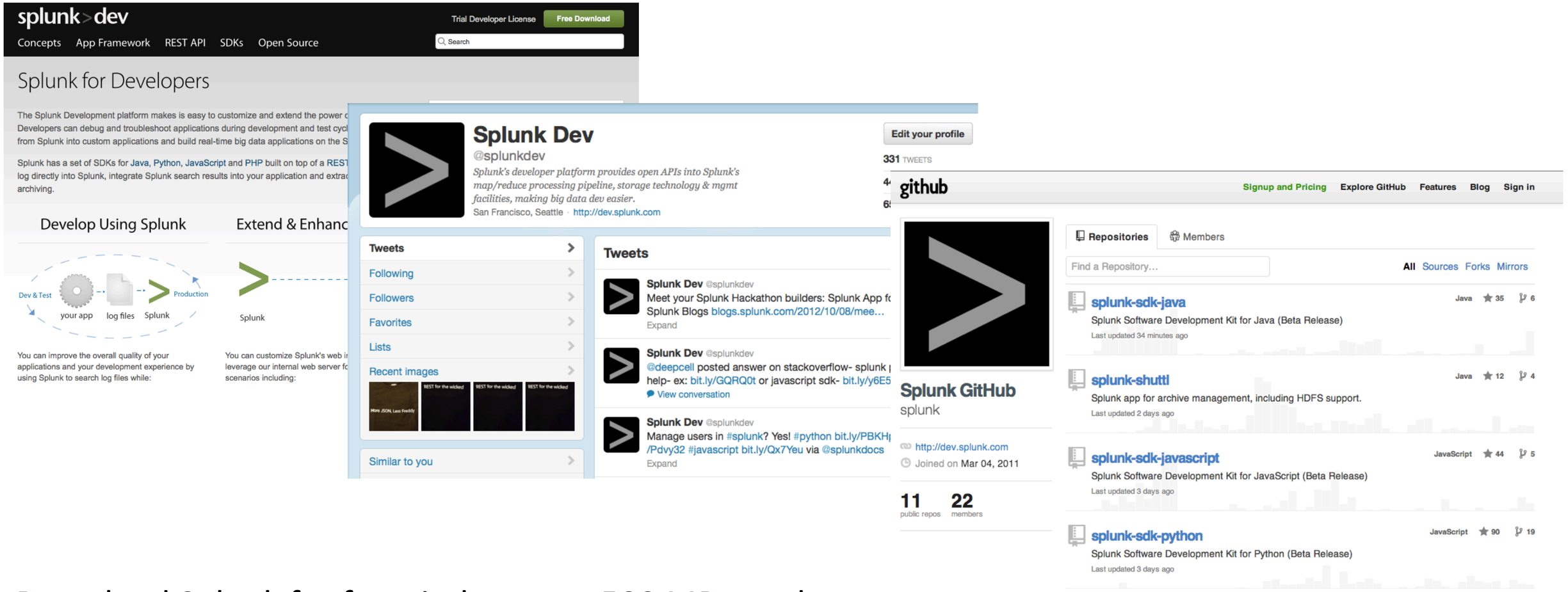






Wrap up

The Splunk Developer Community



- Download Splunk for free, index up to 500 MB per day
- Need more data volume ? Sign up for a 10GB Developer Trial License

Where to Go for More Info

- Email
  - devinfo@splunk.com
- Portal
  - http://dev.splunk.com/
- Github
  - https://github.com/splunk/
- Twitter
  - @splunkdev
- Blog
  - http://blogs.splunk.com/dev/
- Demos
  - http://demos.splunk.com

Contact me

Email : ddallimore@splunk.com
Twitter : @damiendallimore
Skype : damien.dallimore
Github : damiendallimore
Splunkbase : damiend
Slideshare : http://www.slideshare.net/damiendallimore
Blogs : http://blogs.splunk.com/dev

Now go back to your caves and find something, my precious.

Thanks for coming !