Application Forensics Framework (AFF)

Research toward the Development of an Open Source End-to-End Testing System for Complex Transactional Systems

ITEA Annual Technology Review 2011

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Acknowledgements and Disclaimer

This project is funded by the Test Resource Management Center (TRMC) Test and Evaluation/Science & Technology (T&E/S&T) Program through the U.S. Army Program Executive Office for Simulation, Training and Instrumentation (PEO STRI) under contract number W900KK-10-C-0026.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Test Resource Management Center (TRMC) Test and Evaluation/Science & Technology (T&E/S&T) Program and/or the U.S. Army Program Executive Office for Simulation, Training & Instrumentation (PEO STRI).

Avenue Open is a Subcontractor to Venado Technologies, the Prime Contractor for the award.

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Agenda

- Background – Problem Statement
- Project Overview
- Individual Components of the System
- “Demo” in Slide form
- Future Directions
- Acknowledgements and Disclaimer – Project Sponsor

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Background

- **Long-term Trend – Deliver automated segments of Mission Threads as Net-Centric (SOA) Segments**
- **These Net-Centric Segments are typically very complex:**
  - Automate complicated Processes, often via Irregular Paths
  - Are comprised of various Components, which are made up of many levels of Sub-components, etc.
  - Run across a wide array of Infrastructure - Network, Middleware, Computer Hardware, HTTP Processing Layer, SOAP Processing Layer, Application Servers, Application Code Layer, etc.
- **Problem – Little understanding of what is actually happening when a Segment breaks, is slow, etc.**
  - No Ability to “Follow the Messages” as they move through the Mission Thread
  - Little Understanding of what is happening at each level – Network, Infrastructure, Code, etc.

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

First Generation Solutions

- **Adapted Earlier Testing Approaches**
  - User Interface (UI) Focused with Load Simulators
  - “Test the Inside from the Outside” ; the System Under Test (SUT) is a black box
- **Resource Focused**
  - Collect large amounts of data, perform heavy analysis, to determine how each Resource performs individually
- **Little/No Correlation between Resource Measurements at Every Step**
  - For every step in Mission Thread, Resources (processing time, CPU, etc.) can be measured but there is no way to understand how this is impacted by the Application or other factors
  - **Bottom Line: The Tester understands each Resource in isolation but has little understanding of Causality, how other Resources have impacted that Resource.**

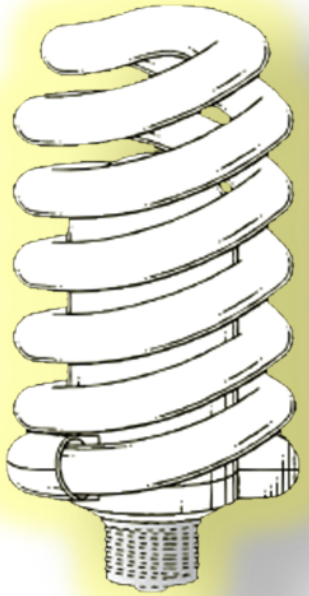
DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Background

- **Testers are Reduced to Guesswork**
  - In some cases, they can find clues in logs
  - Log analysis is “Needle in a Haystack”
- **Poor Results**
  - Extended Testing Times
  - Lower quality feedback to Mission Thread development team
  - Brittle Mission Threads that Break on Battlefield
- **Runtime Issue, so Extends to Production**
  - Testing is just a controlled simulation of Production
  - Issues occur during Runtime, irrespective of scenario, so same issues manifest themselves in Production
  - Solution for Testing arena should be applicable to Production Monitoring as well.

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

A New Way of Thinking….



DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

New Thinking

- **Monitor multiple levels of the OSI stack simultaneously**
- **Focus on collecting correlated data vs. mountains of data**
- **Focus on Actual Transaction Execution – Correlated, end-to-end Transactional Data**

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Full System Vision

- The vision of the AFF system is to discover, follow, and monitor in-depth actual, individual transaction instances end-to-end, across multiple OSI stack layers, while being cognizant of meaningful conditions in the execution environment.
- A transaction instance becomes visible to the AFF from the first network communication made by a requesting device to any given resource, to begin a transaction.
  - A transaction is any request situation – from a simple request over the network to a complex process. It may or may not include a response.
- AFF monitoring will measure and correlate multiple sets of data simultaneously, including network statistics, application servers, OS-level performance data, container-level pre- and post-processing execution details, the thread of application execution data, and message content.
- AFF will reduce TTR by proactively focusing in the primary factors that can affect the execution of any given request

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

AFF Project Description

The Application Forensics Framework (AFF) will develop core components for use in a Net-Centric Application Runtime Testing System.

The End Products will be:

- Reference Implementations of instruments that can be used to collect various types of raw data
- Forensics Frameworks that will enable instant determination of new data collection cut points
- Prototype of a Correlation and Root Cause Analysis system
- Prototype of a Data Dissemination architecture

The project will ultimately target use of resultant technologies as a potential basis for Open Source.

These components will be useful to the Test and Evaluation Community, for any test that includes Net-Centric Services (and potentially other application components as well).

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

AFF Component Breakdown

Various system components have been designed to fulfill the vision

- A **Network Reflector** is an out-of-band, stand-alone server that is attached to one or more span ports, tap devices, or offline Packet Capture (PCAP) sources. It discovers network resources, inter-component relationships, and network latency times, and provides critical information regarding application layout mis-configurations and network issues affecting transaction execution.
- A **Collector** is an in-container monitoring framework for instrumentation of various layers of in-container execution, which provides vital information about the system resources that are invoked at each and every step of a transaction. This includes execution details, execution times, errors and content.
- The **Correlation Mechanism** is a distributed, multi-level system for resource efficient, in-flight transaction correlation. Will correlate data collected at various levels to produce interesting data intersections.

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

AFF After One Year of Research

- **A fully operational reference implementation of a Network Reflector**
  - 1 Reflector with 6 Decoders and Reckoners
- **A fully operational reference implementation of a Collector (Java)**
  - 1 Collector with 9 Monitors
- **Data Collection API - Completed Design**
- **A fully operational Correlation Mechanism**
  - Very Early Stage Prototype, Correlation and Root Cause Analysis Capability

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Decoders for Network Reflector

- Ethernet decoder (technical)
- TCP decoder
- HTTP decoder
- SOAP decoder
- Postgress decoder
- MySQL decoder
- Network latency reckoners for all decoders above, which provides latency evaluation

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Monitors for Java Platform Collector

- Incoming network traffic monitor based on server socket interception technology (currently supports plain HTTP and SOAP over HTTP)
- Outgoing network traffic monitor based on client socket modification technology (currently supports HTTP calls, MySQL and Postgress)
- Monitor for a single thread of an application execution in Java
- Object level monitor for the JVM
- Pre-processing monitor for Tomcat/JBoss
- Post-processing monitor for Tomcat/JBoss
- Environment monitor
- Listener/injector module for network-level transaction correlation purposes
- Shared Memory-based inter-layer correlation mechanism for container level correlation

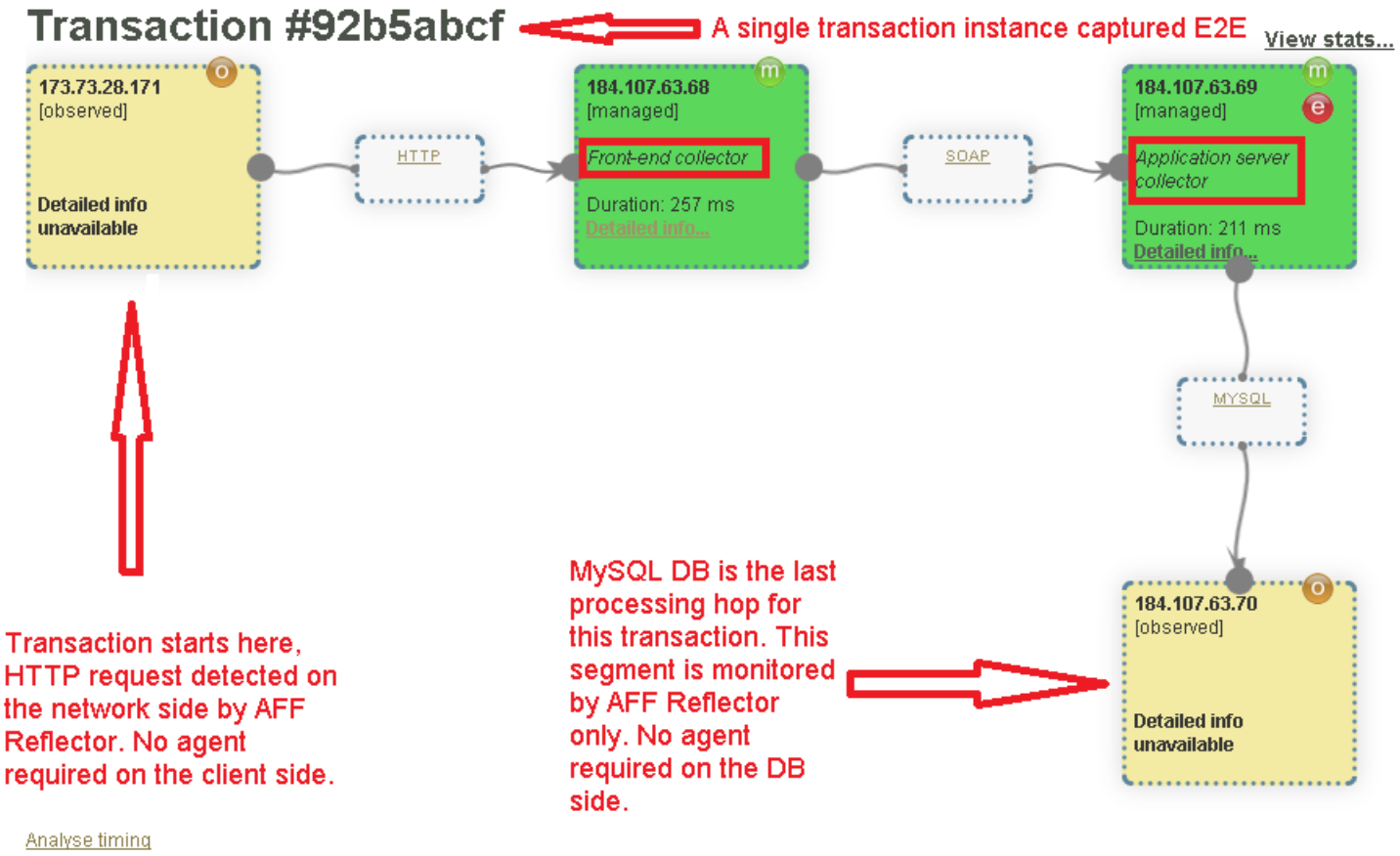
DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Correlation Mechanism

- HTTP content-based correlation – works well for HTTP proxy situations, best suited for non-encrypted HTTP traffic
- TCP marker-based correlation – network low-level, works in most of situations
- TCP marker-less correlation – unique patent-pending technology, best suitable for DB calls correlation, very low performance impact
- All the three correlation techniques used in a combination cover most of existing network layouts

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Automatic End-to-End Transaction Detection and Monitoring



DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Detailed Processing and Content Information Available from the Network

Transaction segment details

Request		Response	
Client	173.73.28.171:1495		
Protocol	HTTP	Server	184.107.63.68:8080
URL	/frontend/sql.do	Protocol	HTTP
Query String		URL	
Transaction Segment Class	Web-app: http://184.107.63.68:8080/frontend/sql.do	Query String	
		Transaction Segment Class	Not Appliable
Method		Method	
Start time	05/19/2011 12:24:18.393	Start time	05/19/2011 12:24:18.650
Managed	false	Managed	false
Error	false	Error	false
Error Msg.		Error Msg.	
Transaction	#92b5abcf	Transaction	#92b5abcf
Content type		Content type	
Content length	147 Bytes	Content length	2612 Bytes
Other parameters		**Other parameters**	
contextID	1665	contextID	1665
httpMethod	POST	Latency (from RTT)	13.434
Latency (from RTT)	19.722	httpCode	200
sessionID	3359	sessionID	3359
transmissionID	29494	transmissionID	29944
httpVersion	HTTP/1.1	chunks	9
chunks	1		

Content:

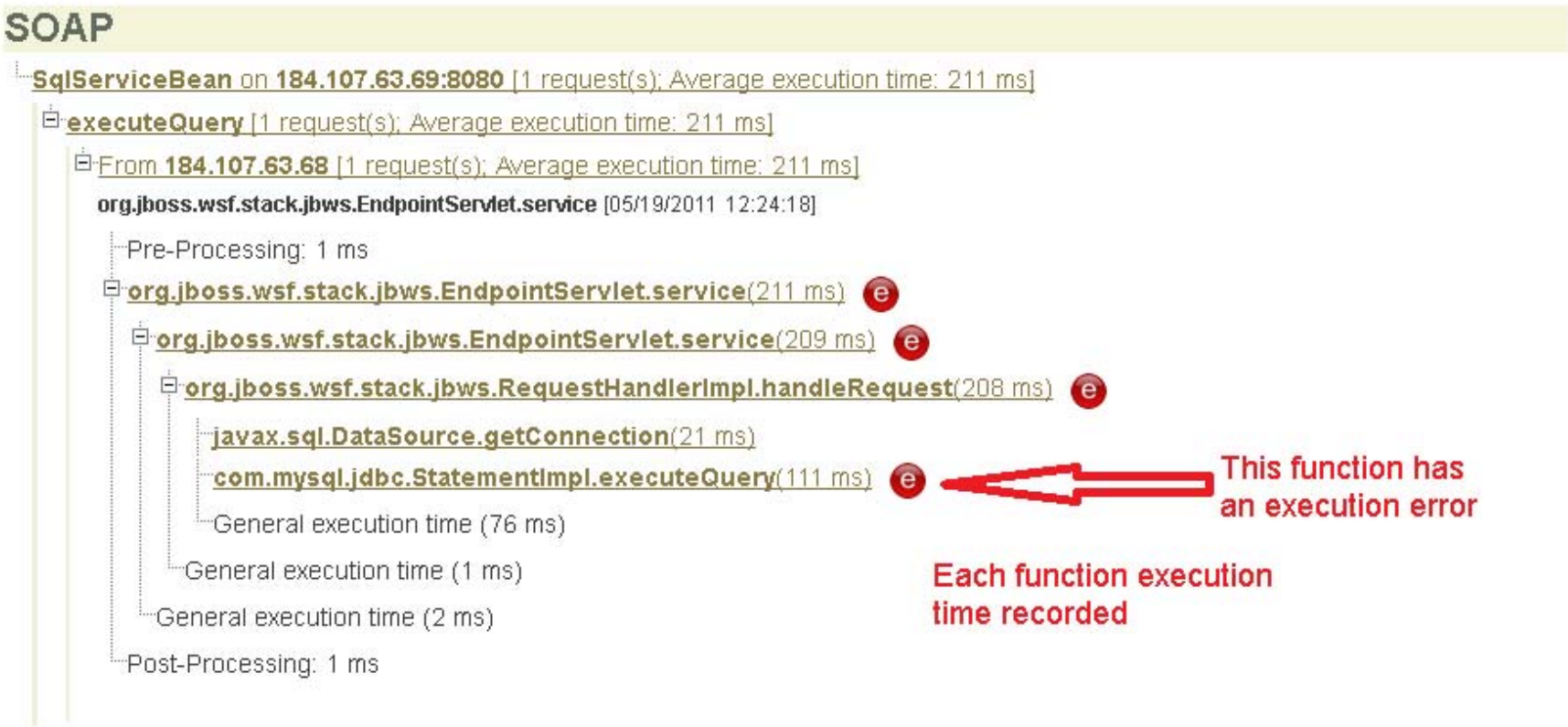
```
rq=3&wsdl=http%3A%2F%2Fcl-a261-020cl.privatedns.com%3A8080%2Faff%2FSqlServiceBean%3Fwsdl&sql=select+lastname+from+employees%3B&counter=3&delay=500
```

Content:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
        <title>AFF Management - Frontend Client</title>
        <link type="text/css" rel="stylesheet" media="all" href="/frontend
```

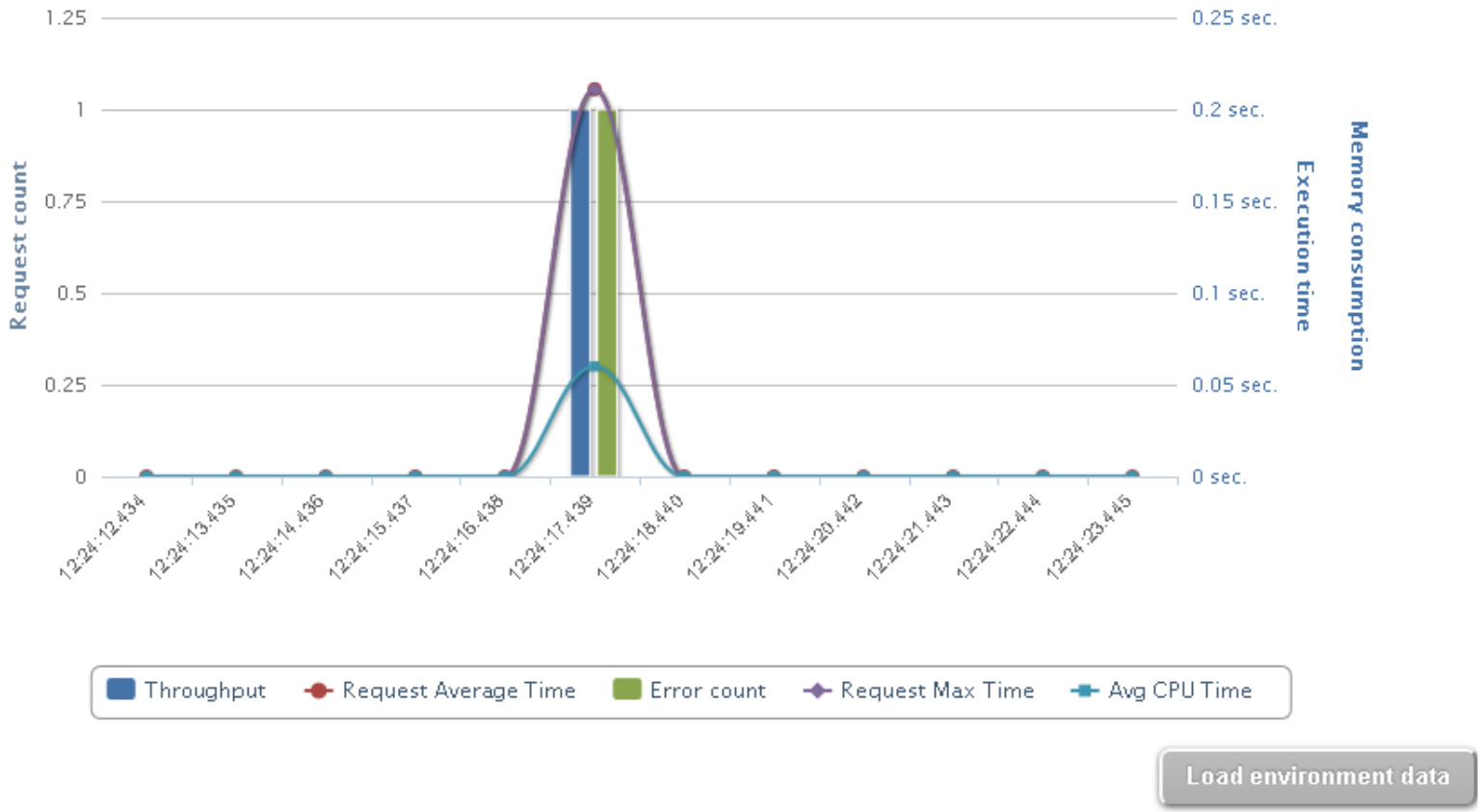
DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Execution Stack Data available from the Monitored Container



DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Resource Consumption Record Eases Task of Analysis



DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Database Performance Monitoring via the Network - Convenient and Powerful

Transaction segment details

Request		Response
Client	184.107.63.69:44644	NO RESPONSE!
Protocol	MYSQL	
URL		
Query String		
Transaction Segment Class	Other: null	
Method		
Start time	05/19/2011 12:24:18.501	
Managed	false	
Error	false	
Error Msg.		
Transaction	Transaction marked as no_opt	
Content type		
Content length	37 Bytes	
Other parameters		
contextID	1690	
Real latency	0.0000	
sessionID	3409	
transmissionID	29490	
chunks	1	

Content:	Content:
!....select lastname from employkees;	NO CONTENT

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

AFF Roadmap

- Revise Java Collector as an extendable application server monitoring framework with the ability to define as many cut points as required for in-depth application forensics. EJB3, asynchronous calls, and more outgoing channels to be supported
- Develop Collector for .Net
- Network Reflector to be extended to support more network protocols, secure communications and determine additional environmental influences on transaction execution
- Develop Policy and Communication Framework – Provide system control and management
- Add Analytical capabilities and supporting UI
- Work with JITC – Develop Use Cases to be run on DCGS Testbed
- Long-term Goal – Mature all System Components

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Project Breakthroughs

- **Success at pioneering Transaction-based analytics vs. Resource-based analytics.**
  - Broader level of information and is much better suited for root cause analysis
  - Original theories floated around this concept but we didn't understand enough to coalesce into a coherent model

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Project Breakthroughs

- **Determined how to develop a single Collector architecture with various technology monitors**
  - Originally assumed the need for multiple Collectors, one per container
  - Can now monitor multiple technologies using same base Collector
- **Determined how to develop a single Reflector architecture with various decoders**
  - Originally assumed the need for multiple Reflectors
  - Have developed multiple decoders and reckoners

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Project Breakthroughs

- **Correlation Moved to Agents**
  - Originally assumed the need for a separate Server
  - New approach significantly more efficient and accurate

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.

Summary

- **AFF Monitors multiple levels of the OSI stack simultaneously**
- **Focuses on collecting correlated data vs. mountains of data**
- **Focuses on Actual Transaction Execution – Correlated, end-to-end Transactional Data**
- **Pioneering Transaction-based analytics vs. Resource-based analytics**

DISTRIBUTION STATEMENT A. Approved for Public Release; distribution is unlimited.