

SATRUN 2014

Identifying and Protecting Architecturally Significant Code

Software Archeology



Mehdi Mirakhorli, Jane Cleland-Huang
DePaul University

Contact me: mehdi@cs.DePaul.edu

Architectural Failures

One Illinois hospital jointly managed by the Departments of Veterans Affairs (“VA”) and Defense (“DOD”) failed to achieve ‘interoperability’ between the Departments’ EHR systems, costing the hospital at least \$700,000 annually.

This is despite the fact that the DOD and VA have already spent \$100 million to achieve this quality.



Architectural Failures

A few days after the launch of the federal government's Obamacare website, millions of Americans that were looking for information about new health insurance plans were locked out of the system even though the designers of HealthCare.gov endeavored to fix the problem and enhance the availability.

Was it just
availability issue?



Architectural Failures

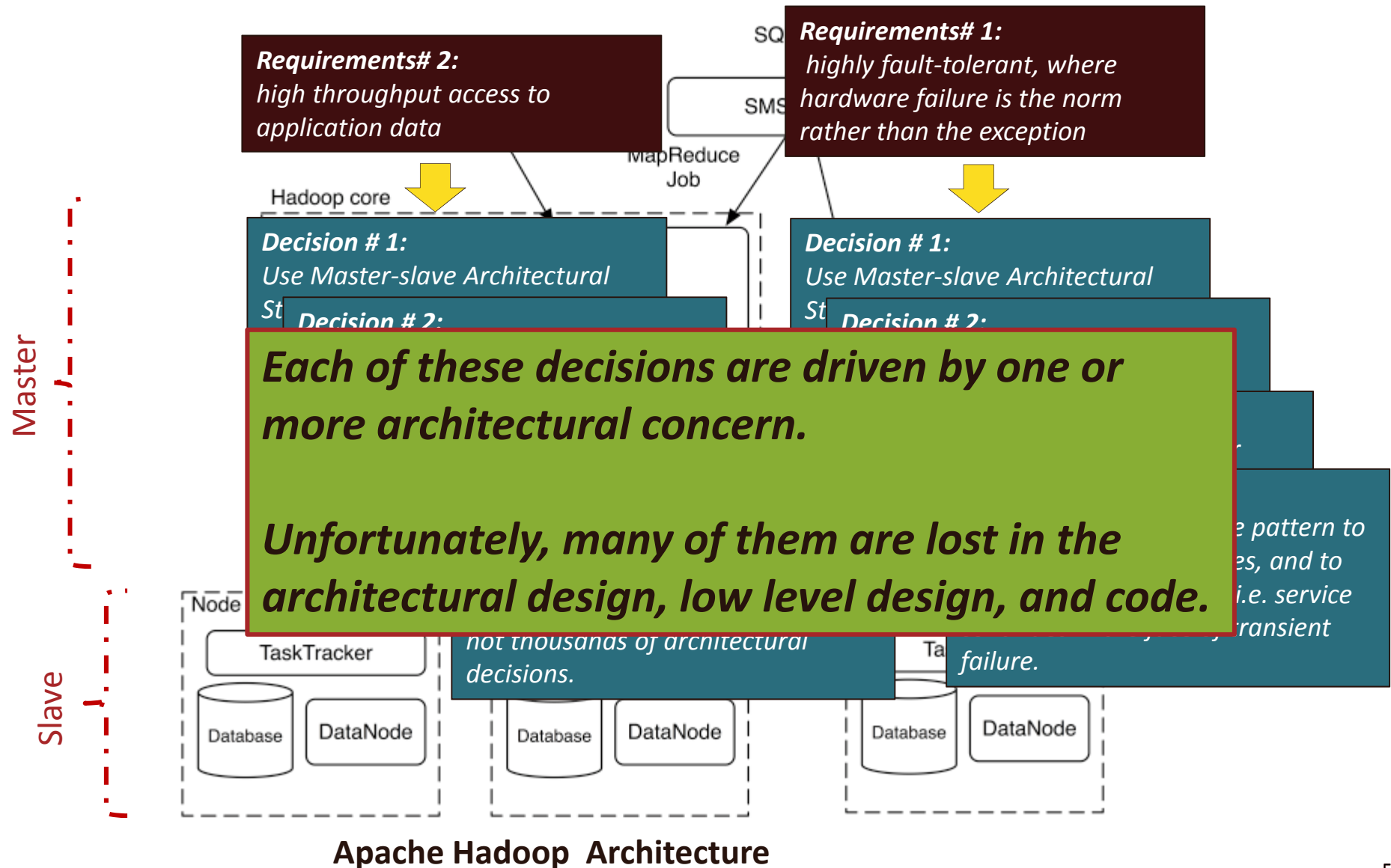


“I identified a series of steps that could be easily automated to collect usernames, password reset codes, security questions, and email addresses from the system -- without any kind of authentication.”

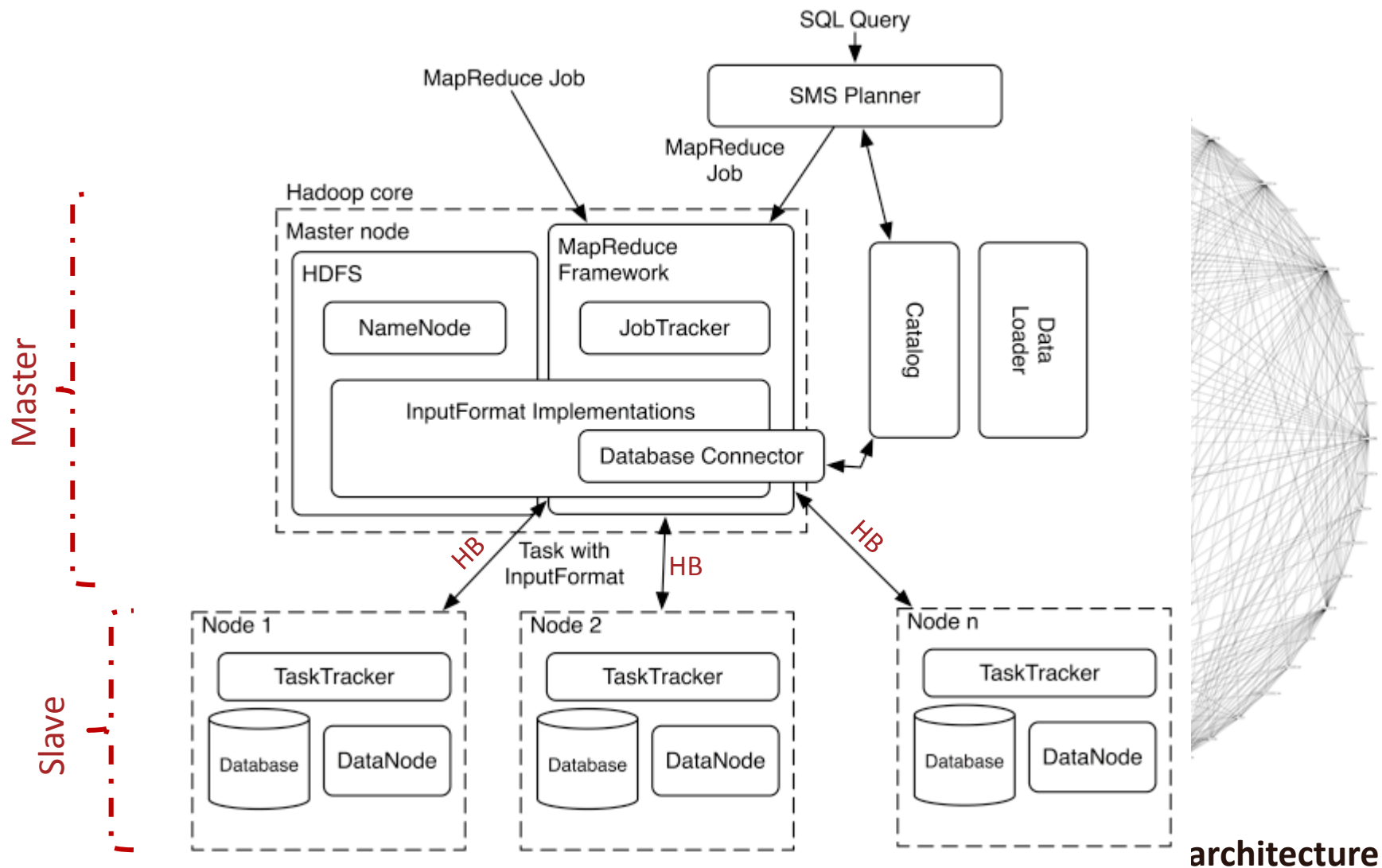
SEBELIUS: **“And we immediately corrected that problem, so there wasn't a -- it was a theoretical problem that was immediately fixed. I would tell you we are storing the minimum amount of data, because we think that's very important. The hub is not a data collector. It is actually using data centers at the IRS, at Homeland Security, at Social Security to verify information, but it stores none of that data, so we don't want to be.....”**



Detailed Example: An architectural view



Detailed Example: Architectural Decay



architecture

Architecture Breaker

Detailed Example in Hadoop:



Developer #1: DataNodes.java, should send several messages to the NameNode.java. Messages such as block reports, heartbeat, blocks to be deleted etc.



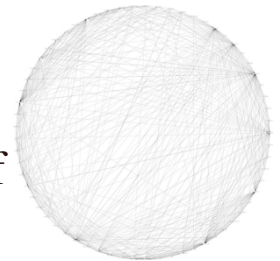
Developer #2: So many messages, lets merge them by piggy-backing
Design Decay & Compromising Availability: block reports are usually delayed, system detects the DataNode failure while it is alive and lunches the recovery process



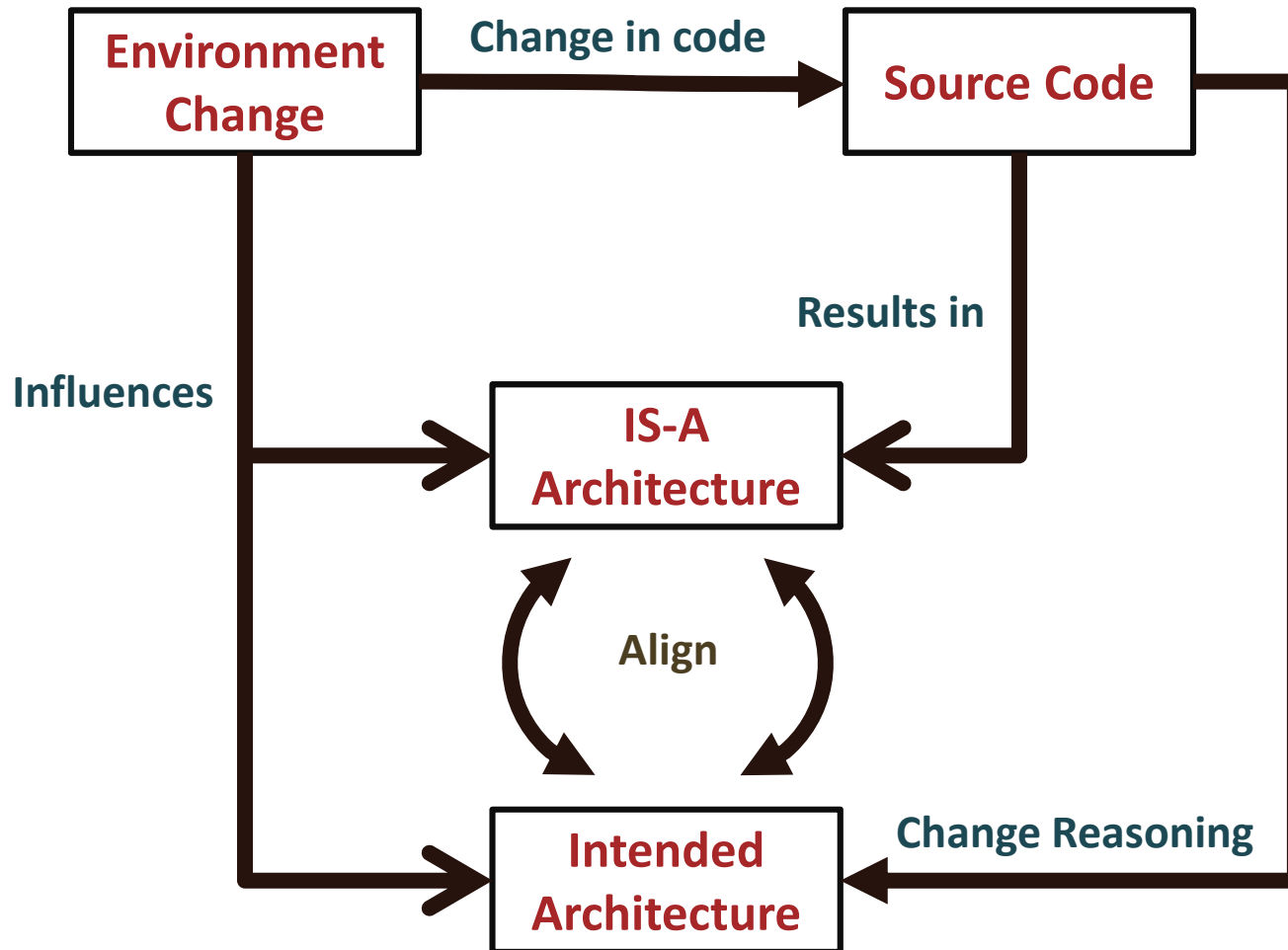
Developer #3: every 10 seconds DataNode reports data or send an empty message for heartbeat



Developer #4: lets make it every 2 seconds
Design Decay & Performance Tradeoff: Performance issues, tradeoff between availability and performance

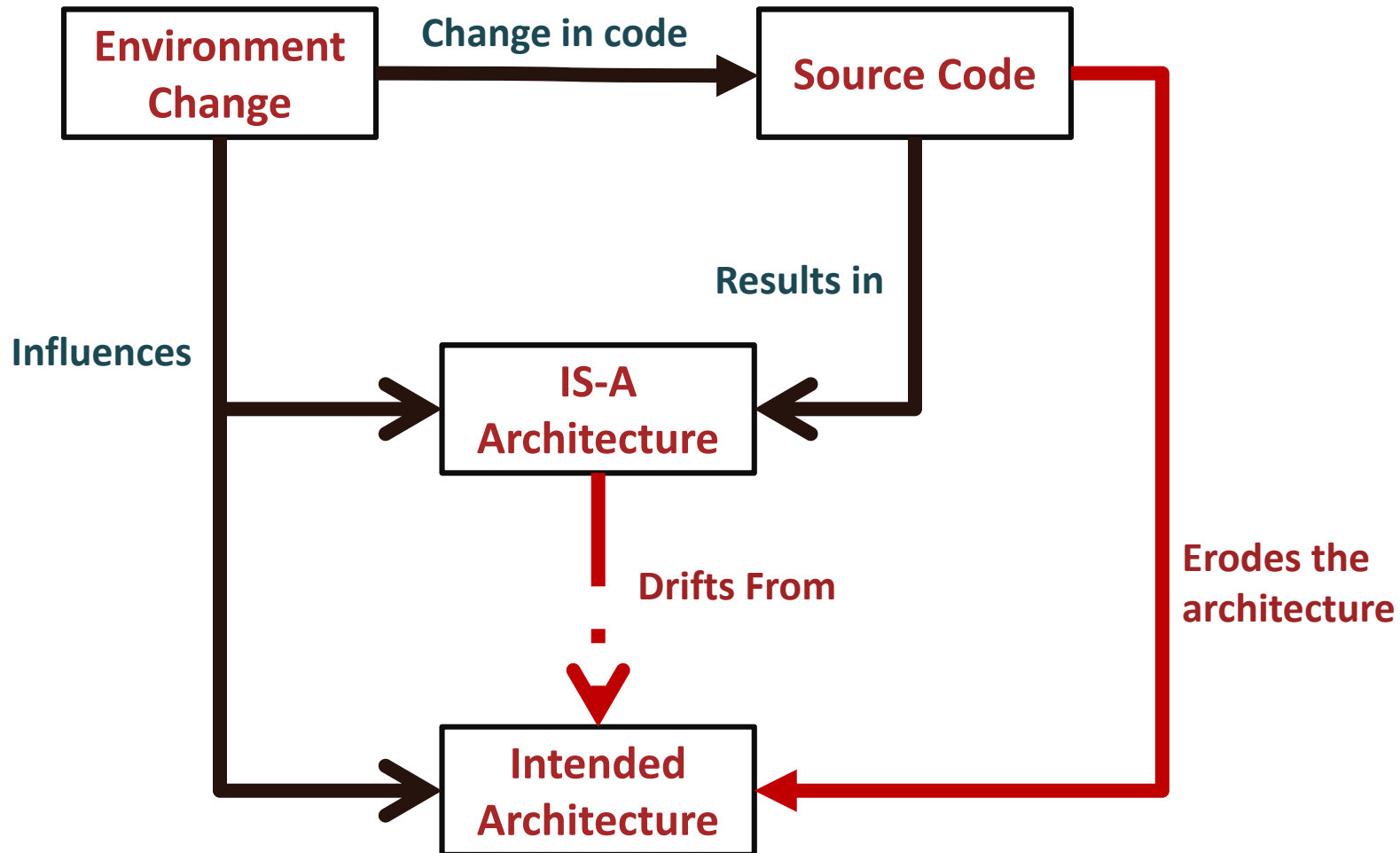


Change Cycle: Ideal World



Ideal World: Architectural information is documented during the Architectural design phase and is updated regularly to reflect the current system architecture.

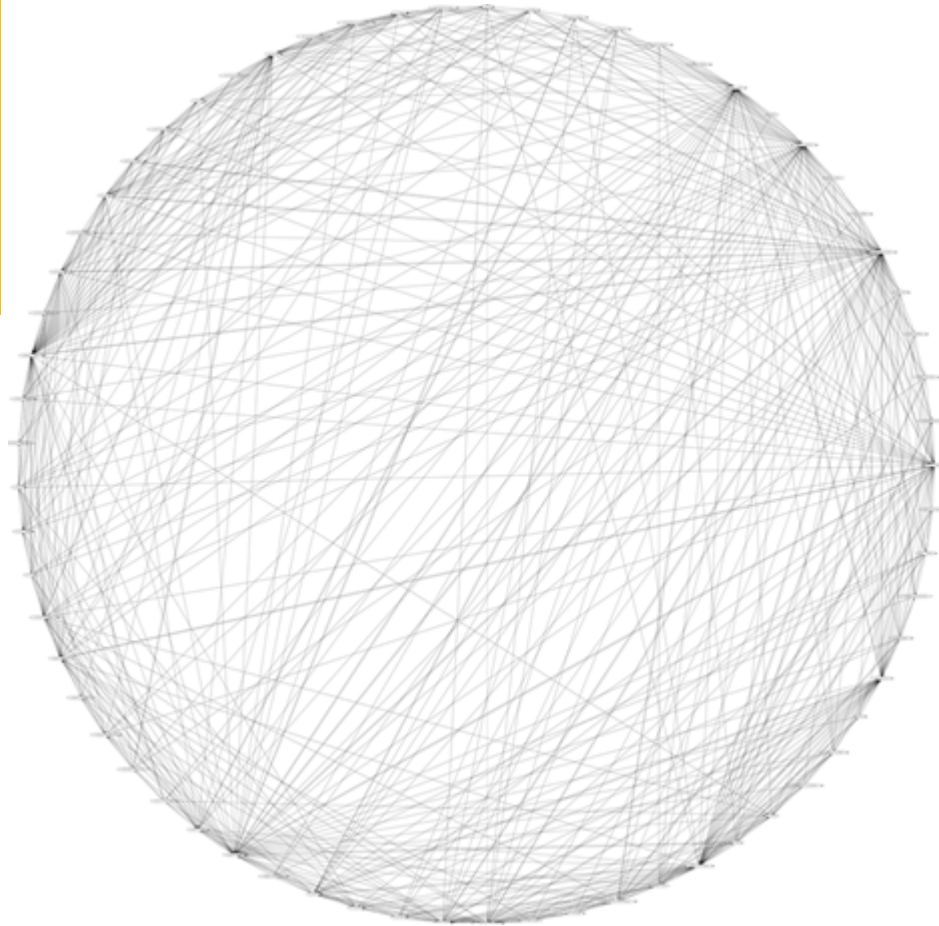
Change Cycle: Real World



Real World: Architectural information is outdated and does not reflect the current architecture of the system.

Architectural Decay

Eroded architecture becomes complex, difficult to understand and difficult to maintain.



A big ball of mud: Apache Hadoop architecture

Archie: A Smart IDE to Protect Architecture



The screenshot displays the Archie IDE interface with four numbered callouts highlighting key features:

- 1 Detection Engine:** Located at the bottom left, it includes a "Scan Directory" field, a "Classification Threshold" slider, and a "Scan" button. Below this is a "TIM Management" section with a list of files.
- 2 Retrieved Code Snippets:** Located at the bottom right, it shows a "Results" section with a "Classification Filter" dropdown and a list of snippets including "LoadBalancing", "PBAC", "Audit", and "Authenticate".
- 3 Code Preview:** Located in the center, it shows a Java code snippet from `FSNamesystem.java` with a red box highlighting the `logAuditEvent` method.
- 4 Visualization:** Located at the top right, it shows a UML diagram with a central "Audit" node connected to "Security", "Privacy", "Rationale", "Audit Manager", and "Audit Logger".

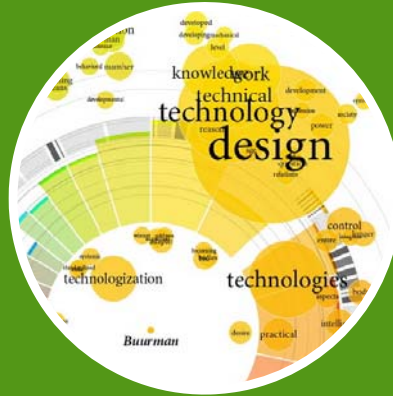
The vision initially presented at:

Mehdi Mirakhorli, Cleland-Huang, "[Using Tactic Traceability Information Models to Reduce the Risk of Architectural Degradation during System Maintenance](#)", *ICSM 2011*.

Archie: A Smart IDE to Protect Architecture



Detect and monitor code snippets that implement key architectural decisions in the source code.



Proactively keep developers informed of underlying architectural decisions during maintenance activities.



Automatically trace external architecture specification documents to the source code or design model.



Perform change impact analysis of architectural concerns at both the code and design level.



Archie: A Smart IDE to Protect Architecture



Detect and monitor code snippets that implement key architectural decisions in the source code.

Decision Detector: A rigorously validated automated technique based on a combination of machine learning, structural analysis, and pattern matching techniques.

Why it works?: Trained by sample source codes of hundreds open source projects.



Code Snippets

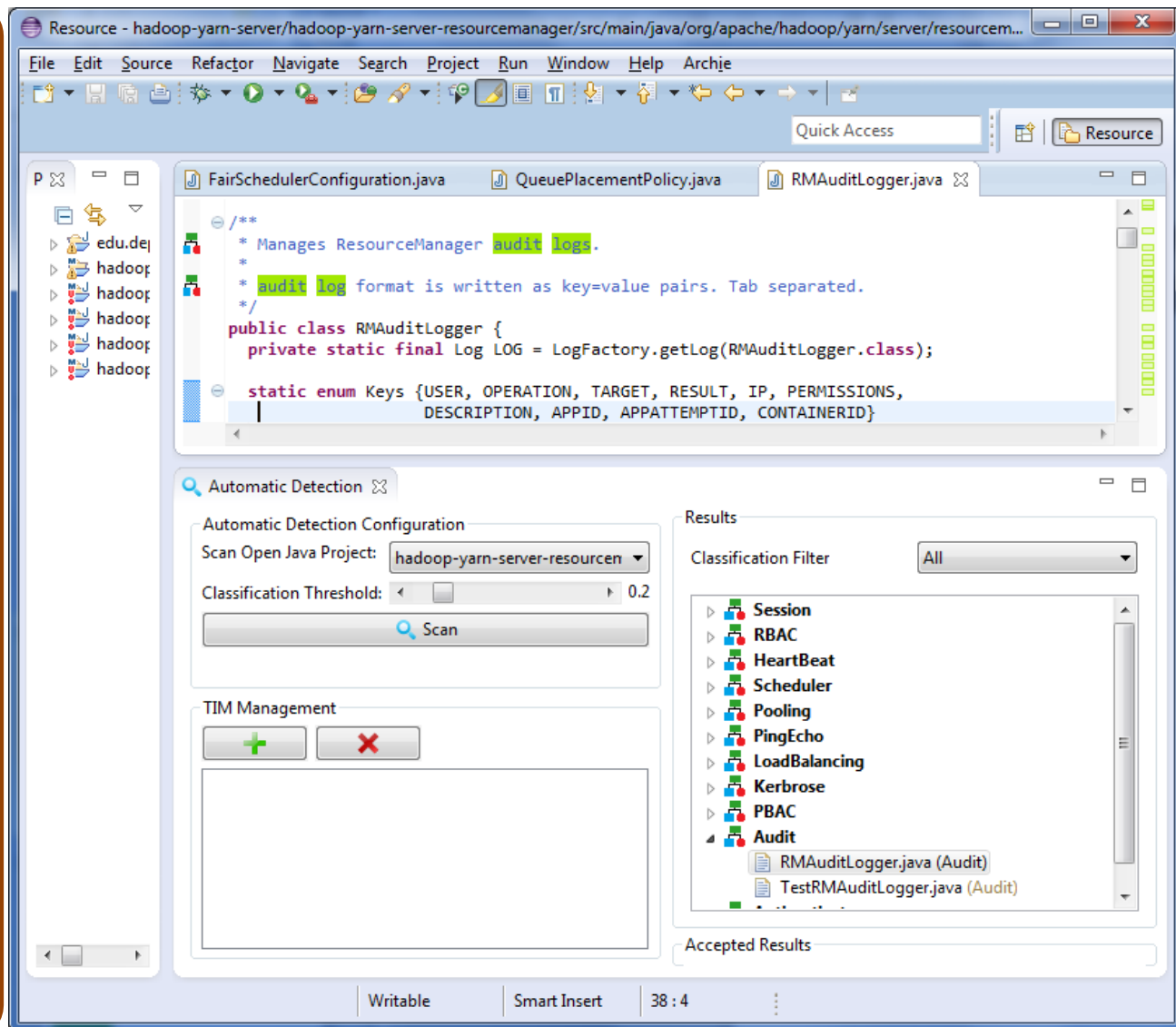
```
public boolean isAuditUserIdentifyPresent() {
    return(this.auditUserIdentify != null);
}

public BigDecimal getAuditSequenceNumber() {
    return(this.auditSequenceNumber);
}
```


Archie: A Smart IDE to Protect Architecture



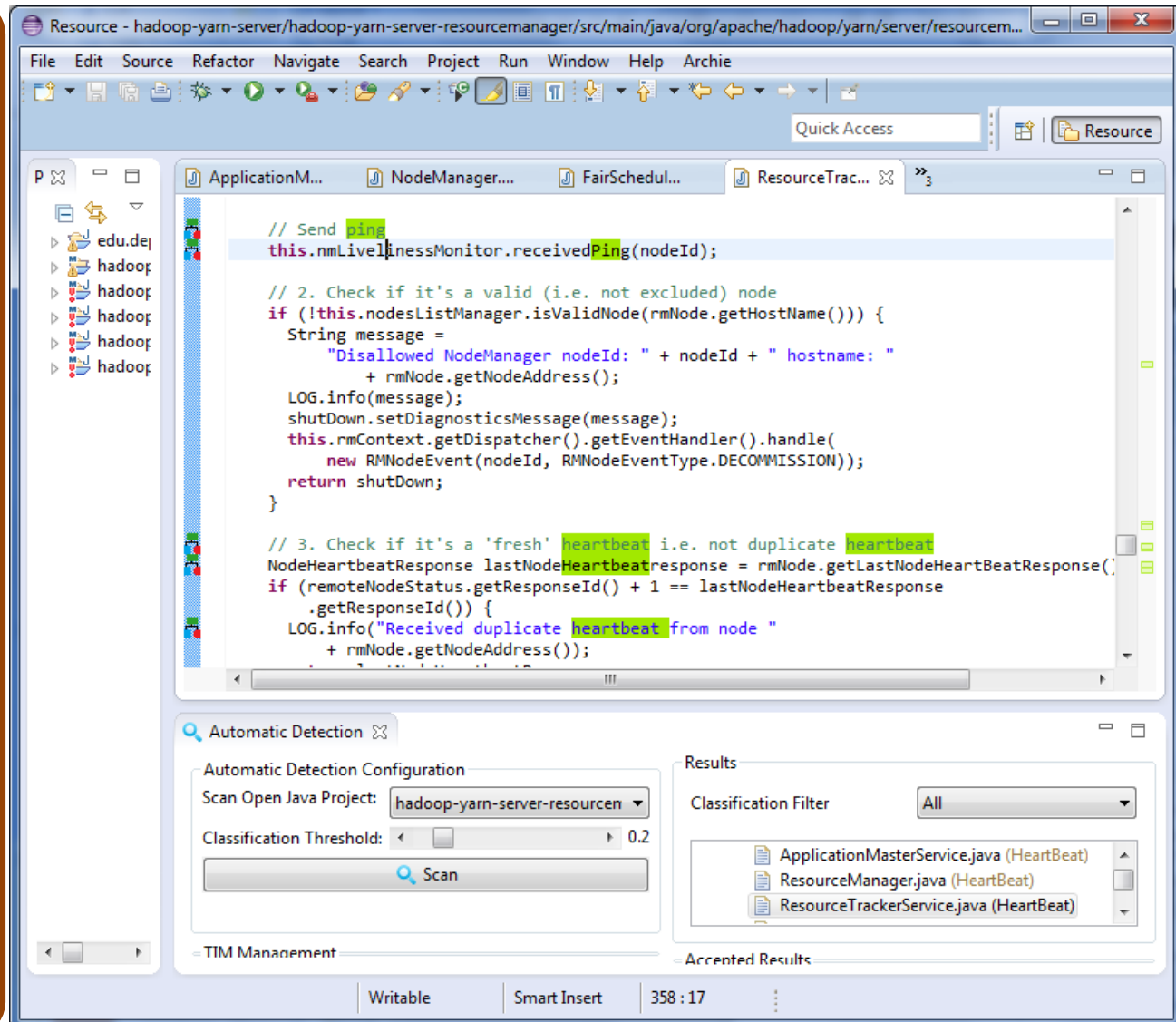
Detect and monitor code snippets that implement key architectural decisions in the source code.



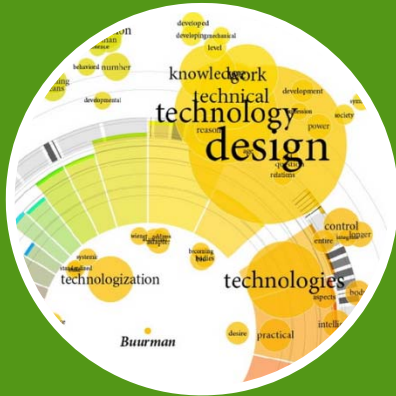
Archie: A Smart IDE to Protect Architecture



Detect and monitor code snippets that implement key architectural decisions in the source code.



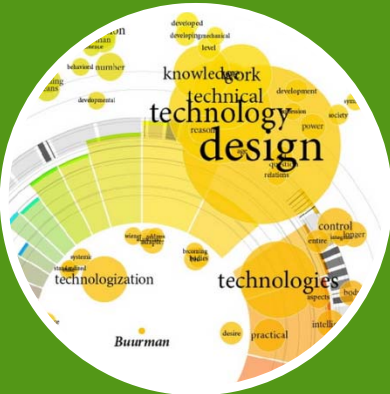
Archie: A Smart IDE to Protect Architecture



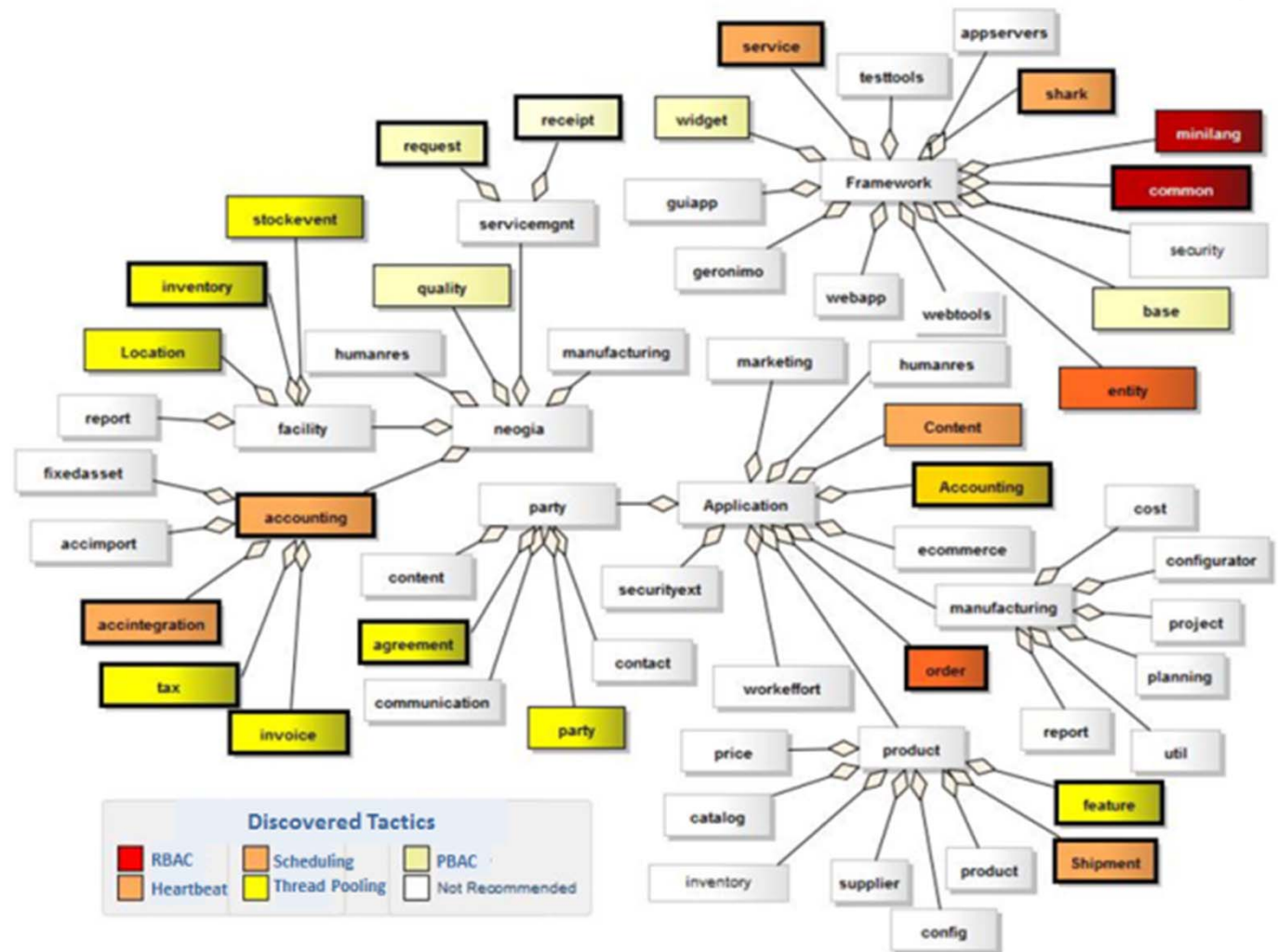
Proactively keep
developers
informed of
underlying
architectural
decisions during
maintenance
activities.

- [illegible]

Archie: A Smart IDE to Protect Architecture



Proactively keep
developers
informed of
underlying
architectural
decisions during
maintenance
activities.



Archie: A Smart IDE to Protect Architecture



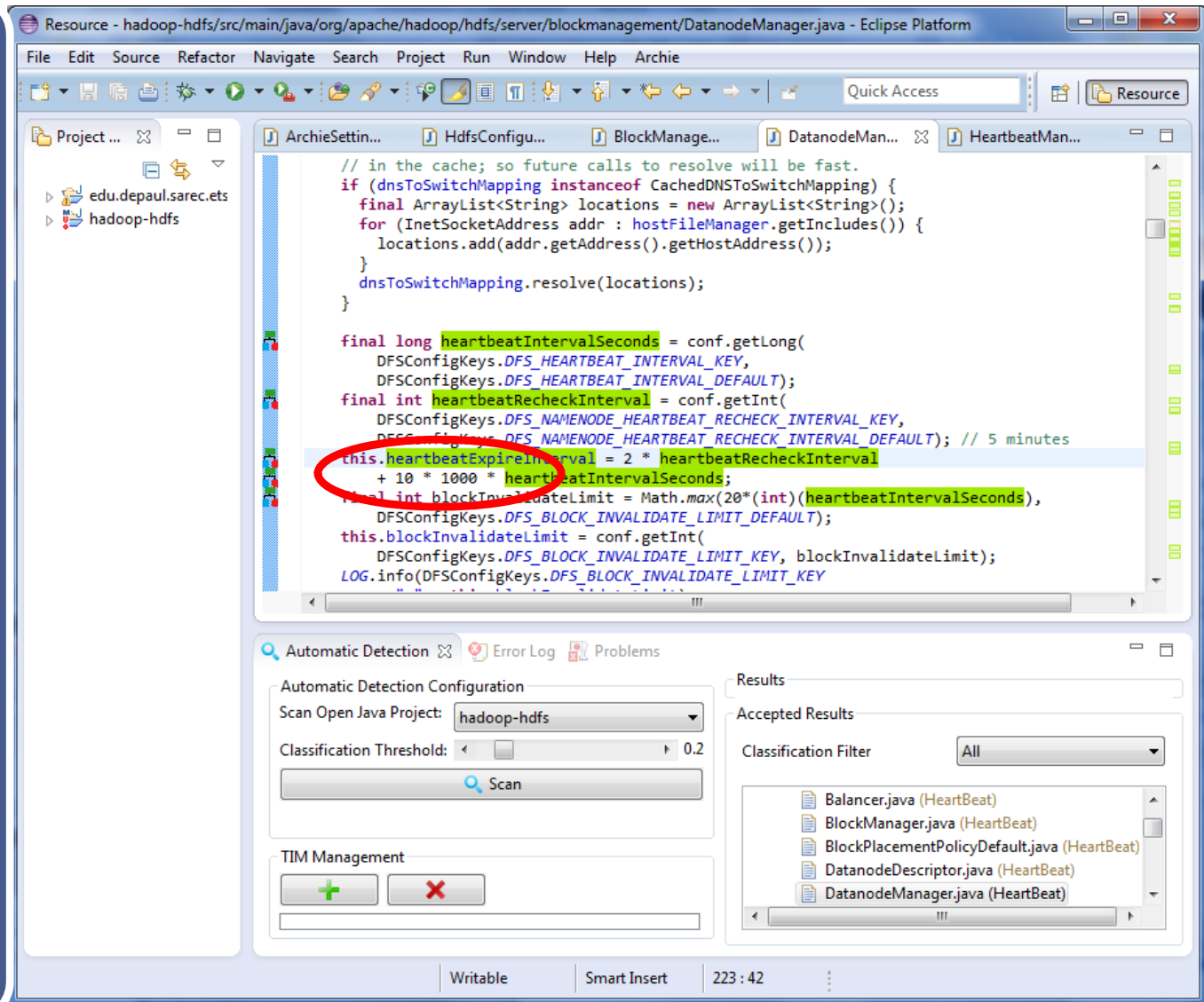
Perform change impact analysis of architectural concerns at both the code and design level.

An asynchronous Event-Based monitoring and notification infrastructure has been designed to proactively inform developers of underlying architectural decisions. An initial proof of concept experiment has been conducted.

Archie: A Smart IDE to Protect Architecture



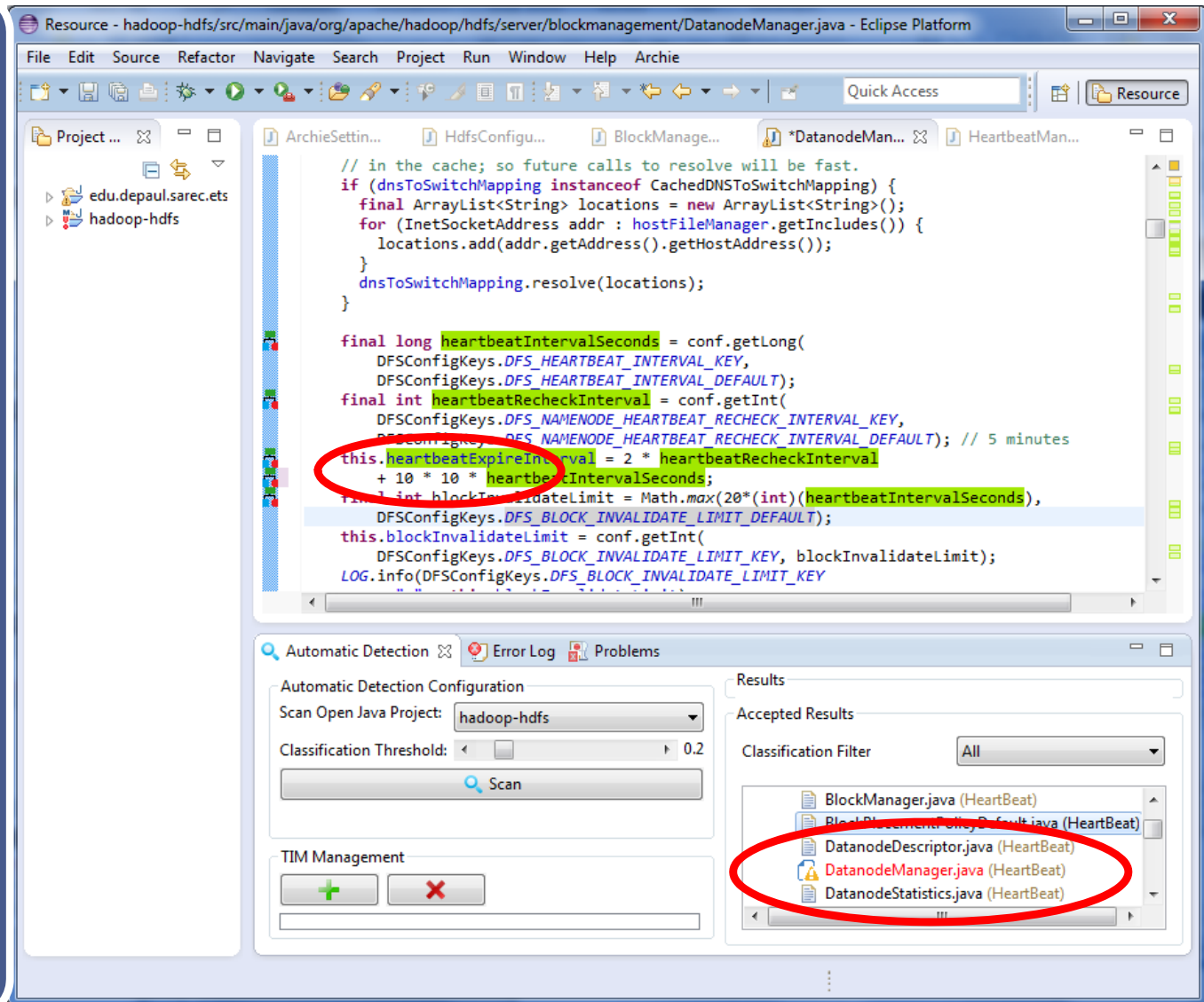
Perform change impact analysis of architectural concerns at both the code and design level.



Archie: A Smart IDE to Protect Architecture



Perform change impact analysis of architectural concerns at both the code and design level.



Archie: A Smart IDE to Protect Architecture



Perform change impact analysis of architectural concerns at both the code and design level.

Resource - hadoop-hdfs/src/main/java/org/apache/hadoop/hdfs/server/blockmanagement/DatanodeManager.java - Eclipse Platform

File Edit Source Refactor Navigate Search Project Run Window Help Archie

Quick Access

Project ...

- edu.depaul.sarec.ets
- hadoop-hdfs

ArchieSettin... HdfsConfigu... BlockManage... *DatanodeMan... HeartbeatMan...

```
// in the cache; so future calls to resolve will be fast.
if (dnsToSwitchMapping instanceof CachedDNSToSwitchMapping) {
    final ArrayList<String> locations = new ArrayList<String>();
    for (InetSocketAddress addr : hostFileManager.getIncludes()) {
        locations.add(addr.getAddress().getHostAddress());
    }
    dnsToSwitchMapping.resolve(locations);
}

final long heartbeatIntervalSeconds = conf.getLong(
    DFSConfigKeys.DFS_HEARTBEAT_INTERVAL_KEY,
    DFSConfigKeys.DFS_HEARTBEAT_INTERVAL_DEFAULT);
final int heartbeatRecheckInterval = conf.getInt(
    DFSConfigKeys.DFS_NAMENODE_HEARTBEAT_RECHECK_INTERVAL_KEY,
    DFSConfigKeys.DFS_NAMENODE_HEARTBEAT_RECHECK_INTERVAL_DEFAULT); // 5 minutes
this.heartbeatExpireInterval = 2 * heartbeatRecheckInterval
    + 10 * 10 * heartbeatIntervalSeconds;
final int blockInvalidateLimit = Math.max(20*(int)(heartbeatIntervalSeconds),
    DFSConfigKeys.DFS_BLOCK_INVALIDATE_LIMIT_DEFAULT);
this.blockInvalidateLimit = conf.getInt(
    DFSConfigKeys.DFS_BLOCK_INVALIDATE_LIMIT_KEY, blockInvalidateLimit);
LOG.info(DFSConfigKeys.DFS_BLOCK_INVALIDATE_LIMIT_KEY, blockInvalidateLimit);
```

Design Warnings

Automatic Detection Error Log Problems

5 errors, 2 warnings, 0 others

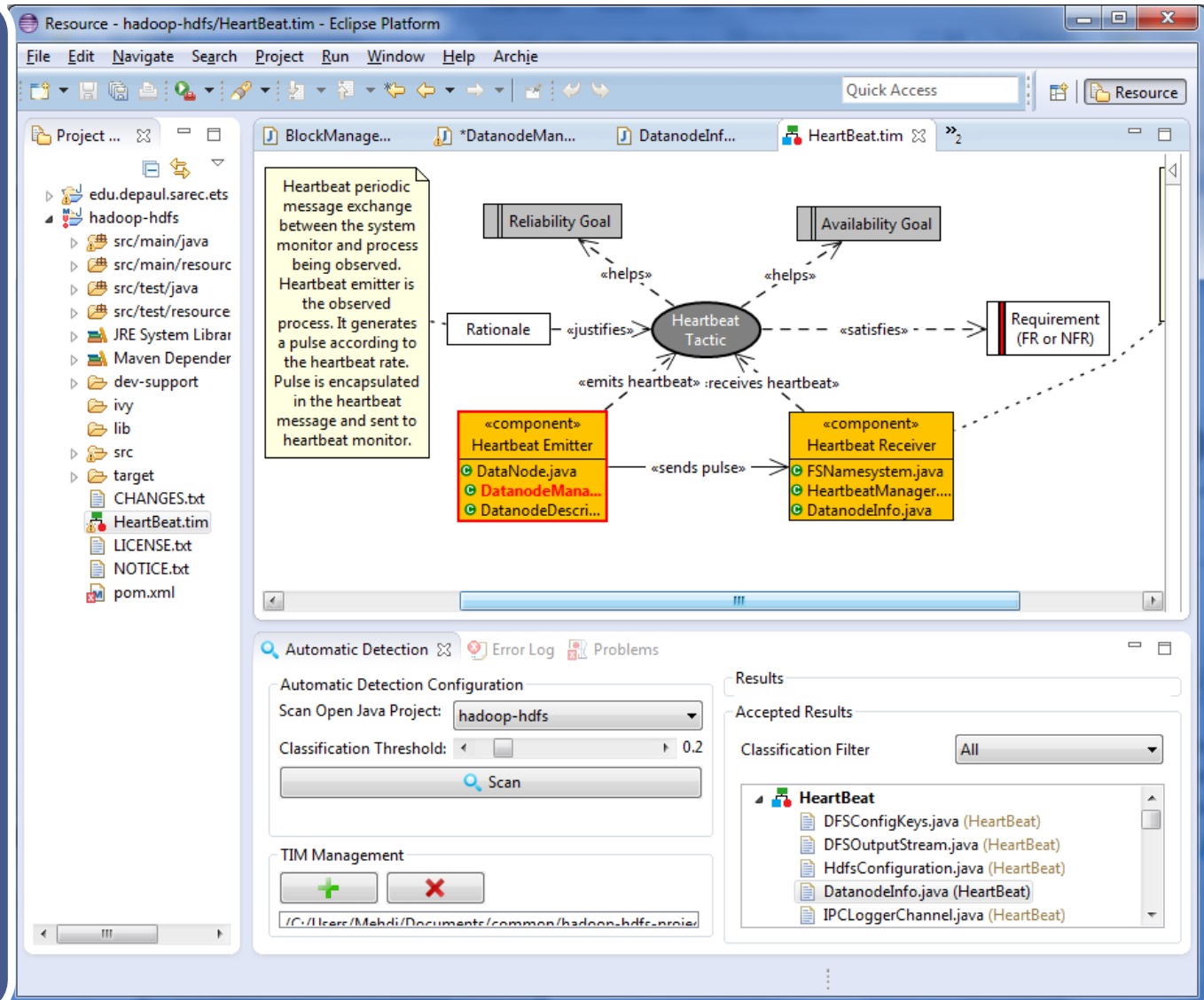
Description	Resource	Path	Location
Errors (5 items)			
Warnings (2 items)			
The file C:\Users\Mehdi\Documents\commo	DatanodeManager.java	/hadoop-hdfs/src/...	line 1
The file C:\Users\Mehdi\Perforce\mmirak_TH	CodeElementEditPart.java	/edu.depaul.sarec.e...	line 1

Warnings (2 items)

Archie: A Smart IDE to Protect Architecture



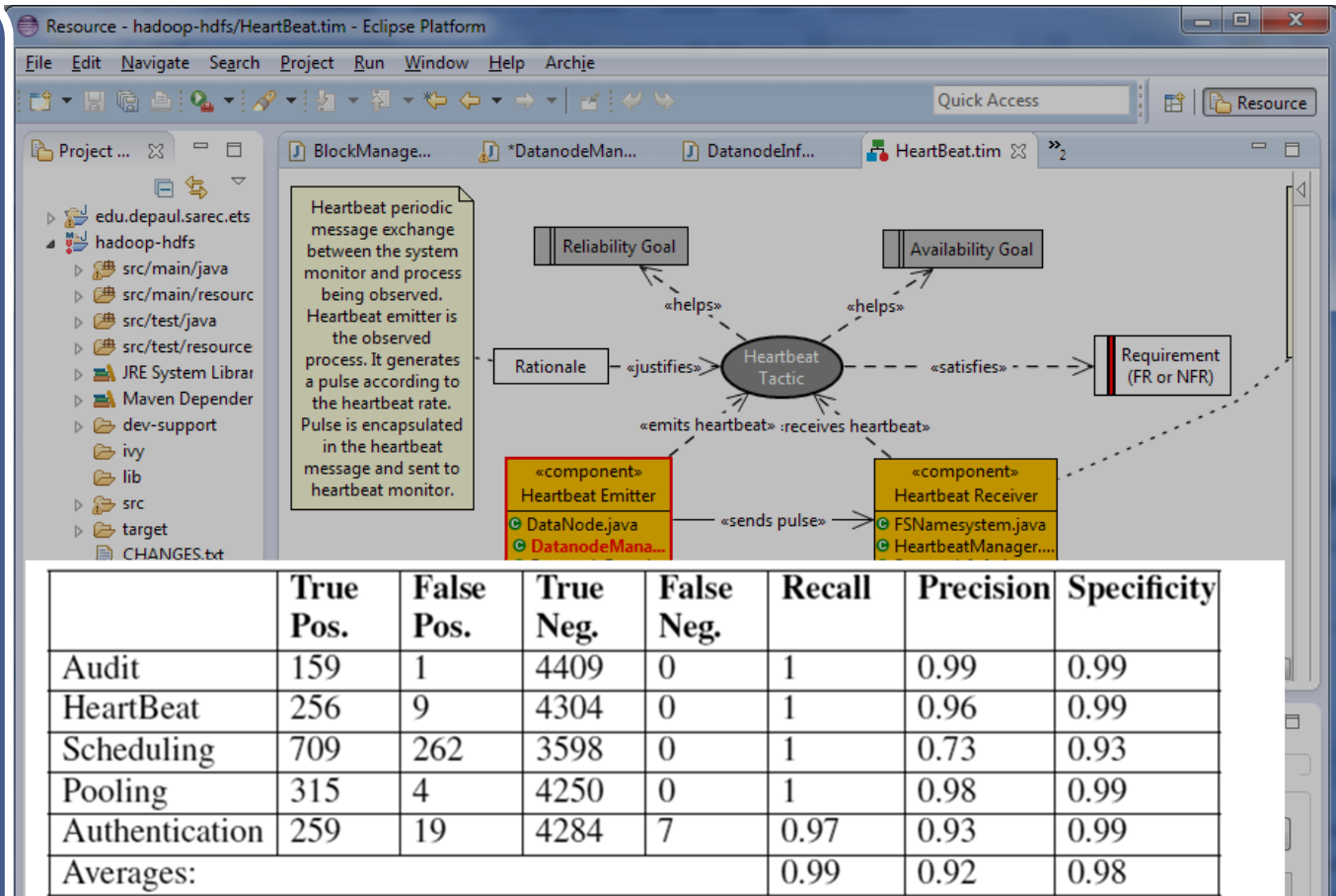
Perform change impact analysis of architectural concerns at both the code and design level.



Archie: A Smart IDE to Protect Architecture



Perform change impact analysis of architectural concerns at both the code and design level.



We utilized the Hadoop change logs for the past four releases, and simulated a change impact analysis scenarios.

Archie: A Smart IDE to Protect Architecture



Automatically
trace external
architecture
specification
documents to
the source code
or design
model.

Poirot : TraceMaker

Project Query Artifacts Options Help

Standard Query > Report

Query: Document ID: 6.9.7
Joints and connections: Gasket materials shall be of either neoprene or other similar material resistant to any action of gas. Natural rubber shall not be used.

<<>> **GasReqsForPaper**

CANDIDATE LINKS UNLIKELY LINKS ID: Find Save

Document ID	Document Description	Confidence Level	Accept
A201	Below Ground Pipe: Buried gas supply pipe shall have non-metallic, flat, ring type, flange gaskets.	■■■■	<input checked="" type="checkbox"/>
A241	Natural Gas Pipe Casings: Neoprene transition end jackets shall be used between pipe under railway tracks and pipe casing.	■■■■	<input type="checkbox"/>
A215	Above Ground Pipe: Above ground gas supply pipe of NPS larger than 50 mm shall have non-metallic, flat, ring type, flange gaskets.	■■■■	<input checked="" type="checkbox"/>
A240	Natural Gas Pipe Casings: Plastic insulating spacers shall be used between casings and neoprene transition end jackets for pipes under railway tracks.	■■■■	<input type="checkbox"/>
A199	Below Ground Pipe	■■■■	<input type="checkbox"/>
A203	Below Ground Pipe: Joints in buried gas supply pipe shall be butt weld connections.	■■■■	<input type="checkbox"/>
A213	Above Ground Pipe	■■■■	<input type="checkbox"/>
A292	Input: The Natural Gas Pipeline network shall use commercial grade natural gas from the Gas Utility Company.	■■■■	<input type="checkbox"/>
A276	Plug type	■■■■	<input type="checkbox"/>
A210	Above Ground Pipe	■■■■	<input type="checkbox"/>

1 - 10 of 52 Next Last

Modify Query - Windows Internet Explo...
http://golevka.cstcis.cti.depaul.edu/Poirot/ArtifactDet

Query modification Enable clouds: ☒

Additional words:
flange

Add word: Add

Click on the underlined term to filter it out

Query:
- Joints and connections
Gasket material Found in be of
either neoprene Query other
similar material resistant
to any action of gas. Natural
rubber shall not be used.

Filter All Clear All Re-run query

Internet 100%

Supporting traceability of distributed heterogeneous software artifacts.

The *Software Assurance Marketplace*

- Archie is integrate into the pool of security tools at SWAMP.
- Will be Integrated with vulnerability analysis tools.

		Has access to all typing and semantic information Tools are language specific.	Sees all the code and the results of optimization and linking. Needs to derive program semantic information from code.	Sees all the code and the results of optimization and linking. Needs to derive program semantic information from code.
		Source	Bytecode	Binary Code
Sees all possible program paths. Over approximates program behavior, i.e., subject to false positives	Static	C/C++/Objective-C: Clang Static Analyzer, Cppcheck, Oink, OCLint, CxSuite Java: PMD, Android Lint Multi-Language: Coverity SAVE, Fortify Static Code Analyzer, CodeSonar, Klocwork Solo	Java: FindBugs, Find Security Bugs Android: Comdroid, SCanDroid, Stowaway, BAP, findbugs-for-android	CodeSurfer/X86, Veracode, BAP, Bugscam/IDAPro
Based on actual program executions; i.e., if a behavior appears, then it is real. Analysis is only as complete at the test input.	Dynamic		Memory: JTest Threading: JTest	Memory: Valgrind, Purify, dmalloc, C/C++ Test, BoundChecker. Threading: Helgrind, DRD, Thread Checker, Parallel Inspector



“We’re trying to do our job in protecting our nation’s critical infrastructure and providing capabilities to be more proactive instead of reactive to cyberthreats. Along with the technologies I’m developing, I think the SWAMP will definitely be a revolutionary force in the software assurance community. We anticipate advancing some breakthroughs in the SWAMP,” Kevin Greene declares.

Kevin E. Greene

Program Manager (SwA), DHS S&T Cyber Security Division (CSD)

The Software Assurance Marketplace


SWAMP

About

Contact

Help

Sign Out



[Home](#)

[My Account](#)

PROJECTS I OWN

Archie's Test

[Add New Project](#)

PACKAGES I OWN

Archie@SWAMP

[Add New Package](#)

DetailsMembersAssessmentsRun RequestsRunsResults

Archie's Test Assessment Runs

Filters

DatePackageToolPlatformLimitReset

The following assessment runs are currently available for project Archie's Test of any package using any tool on any platform limited to 50 items.

Date / Time	Package	Tool	Platform	Status
2014-04-29 07:23	JSPWiki 2.5.139	Archie 1.3	Fedora Linux 19 64-bit	Finished
2014-04-28 14:42	Camel 2.11.1	Archie 1.3	Fedora Linux 18 64-bit	Finished
2014-04-28 14:26	Swamp Java 1.1.0b	Archie 1.3	Ubuntu Linux 12.04 LTS Lucid Lynx 64-bit	Finished
2014-04-28 14:14	NIST Juliet Java CWE600_01 1.2 1.2	Archie 1.3	Red Hat Enterprise Linux RHEL6.4 32-bit	Finished
2014-04-28 14:11	Hadoop 1.1.2	Archie 1.3	Fedora Linux 18 64-bit	Finished
2014-04-22 19:56	Hadoop 1.1.2	Findbugs 2.0.2 (FindSecurityBugs 1.1.0)	Debian Linux 7.0 64-bit	Finished



"All I'm saying is now is the time to develop the technology to deflect an asteroid."

SATRUN 2014

Identifying and Protecting Architecturally Significant Code

Software Archeology



Mehdi Mirakhorli, Jane Cleland-Huang
DePaul University

Contact me: mehdi@cs.DePaul.edu