

Cernunnos BOF

Drew Wills

JA-SIG Conference, March 2, 2009

Dallas, TX

© Copyright Unicon, Inc., 2006. This work is the intellectual property of Unicon, Inc. Permission is granted for this material to be shared for non-commercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of Unicon, Inc. To disseminate otherwise or to republish requires written permission from Unicon, Inc.

Questions

Project History & Cernunnos Basics

???

- How many have looked at Cernunnos?
- What is unclear?
- What blockbuster integrations can we add?

Content

Project History & Cernunnos Basics

Cernunnos at a Glance

Project Home Page:

<http://cernunnos.googlecode.com/>



Discussion Group:

<http://groups.google.com/group/cernunnos-discussion/>

Manual:

<http://cernunnos.googlecode.com/svn/manual/index.html>

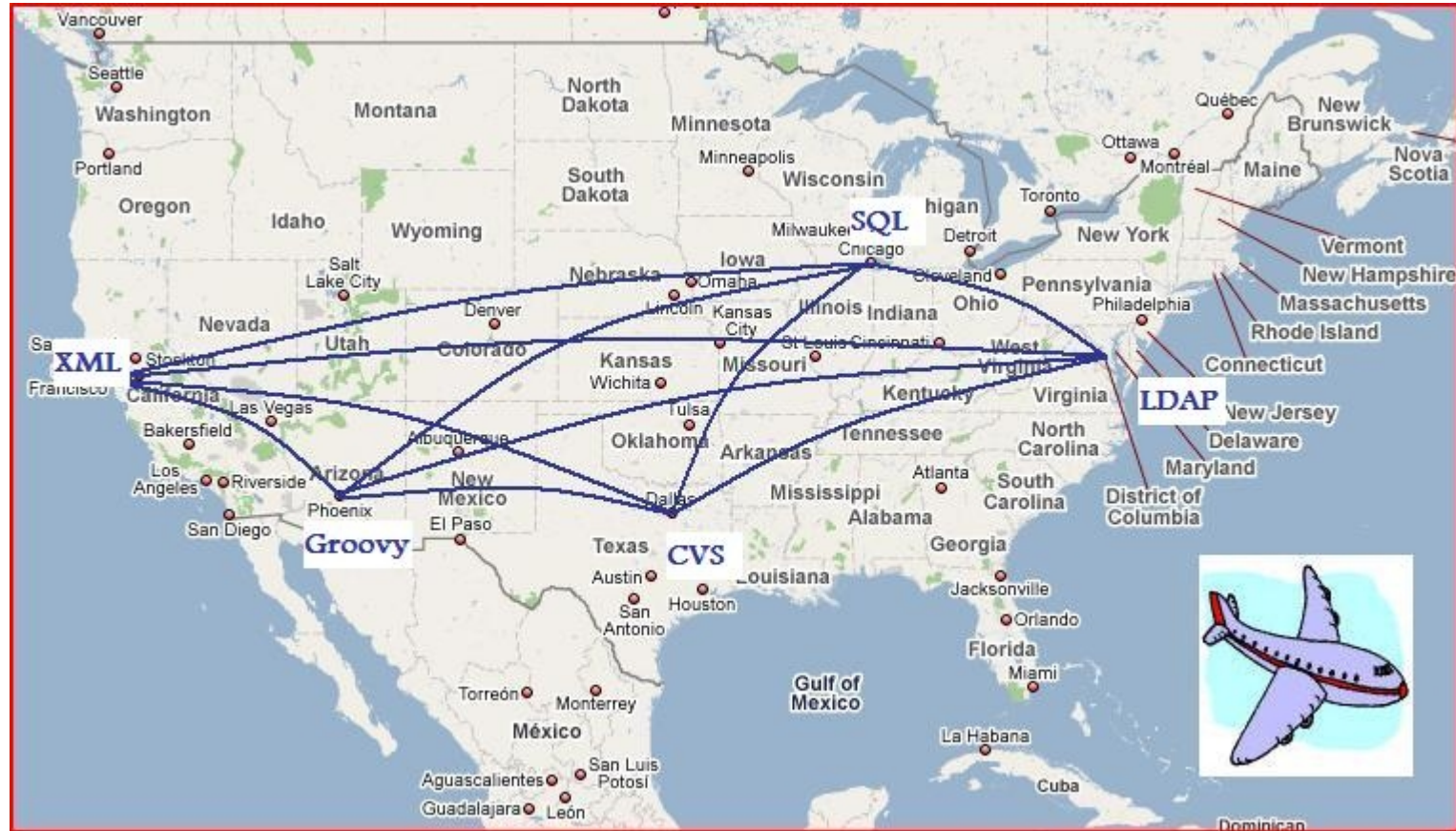
Project Status:

- Version 1.0.0 released September 14th, 2008
- 8 Project Members
- > 20k lines of source (code, comments, blanks)
- > 450 commits since February 2007

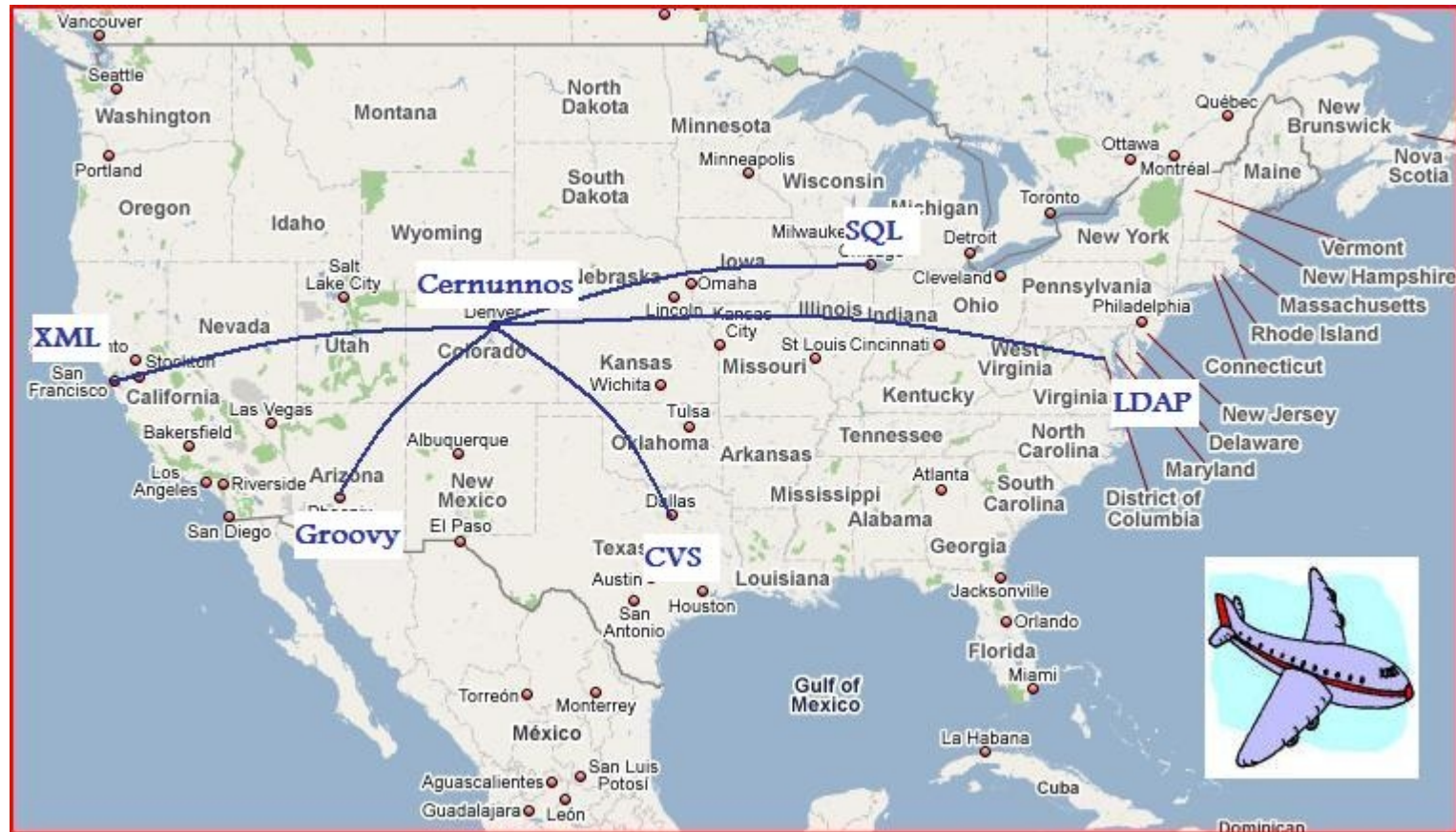
What is Cernunnos?

- Cernunnos helps you be more productive
- *Here's how it works...*
 - You don't have to tell components, subsystems, or objects *how* to work together
 - You just have to tell them to do so
 - This simple difference reduces busywork and bulk dramatically
 - It's like a **hub airport** for code

Welcome to *Java Airways*



Try Cernunnos Airways Instead



Jigsaw Puzzles vs. LEGO Bricks

- Consider another example: **jigsaw puzzles**
 - Puzzle pieces only combine in one way
 - If you want to reuse puzzle pieces, **you have to create new pieces** that will accept their unique shapes
- Each LEGO brick, however, already combines with every other LEGO brick -- past, present, and future
- Cernunnos is like *LEGO-typing for the Java Platform*

GOAL: Combine A & B

$$A + C + B = AB$$



$$A + B = AB$$

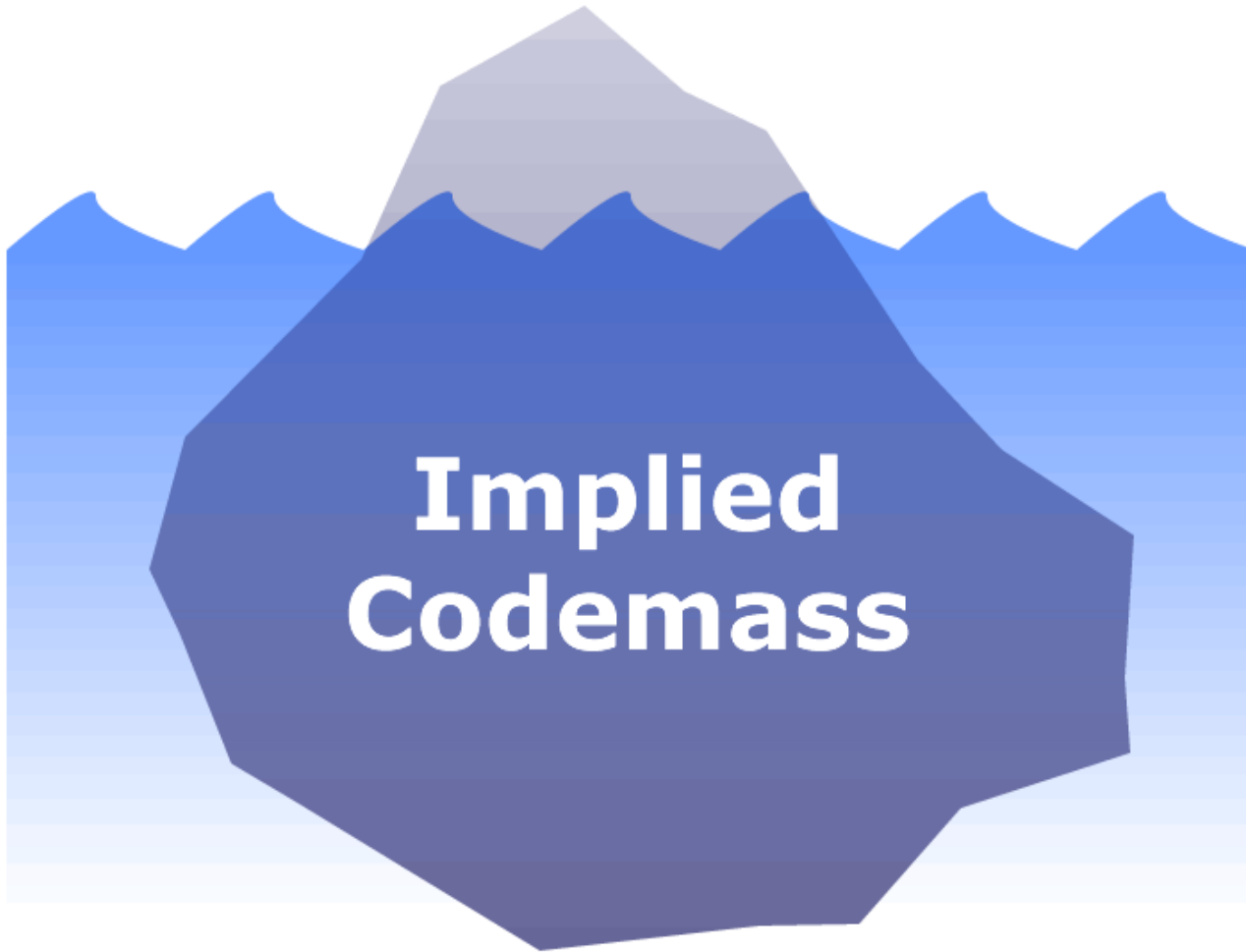


Implied Codemass

- Let's agree the *codemass* of a solution is the sum of all custom code
- *Implied codemass* is the portion of a solution that you're forced to write by architectural choices
- The real bulk, footprint, or mass of an individual choice -- whether you write it or download it -- includes the burden it places on components that interact with it

`innate mass + implied mass = total mass`

Hidden Bulk



**Implied
Codemass**

Example: SQL Query with JDBC

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

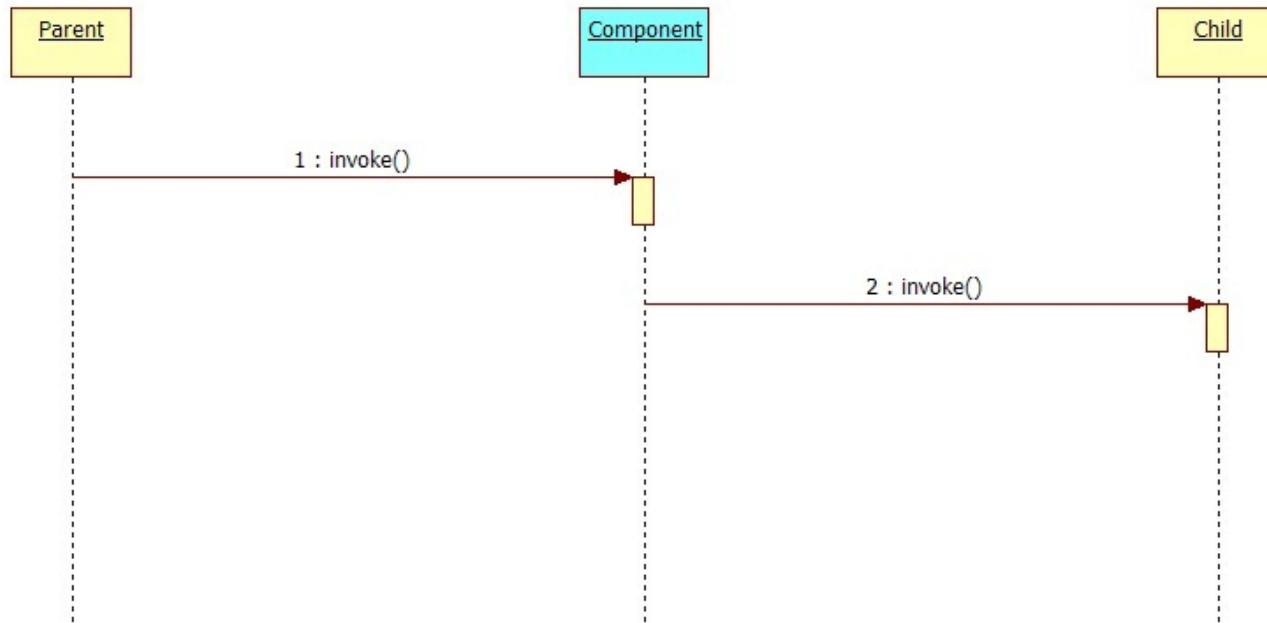
[...]

Connection conn = null;
PreparedStatement pstmt = null;
try {

    Class.forName(sqlDriver);
    conn = DriverManager.getConnection(sqlUrl, sqlUser, sqlPassword);
    pstmt = conn.prepareStatement("SELECT * FROM up_user");
    ResultSet rs = pstmt.executeQuery();
    /* do something useful with the results */

}
catch (ClassNotFoundException cnfe) { /* do something... */ }
catch (SQLException sqle) { /* do something... */ }
finally {
    pstmt.close();
    conn.close();
}
```

Hypothetical Sequence Diagram



Why aren't the previous examples drop-in replacements for Component?

Example: SQL Query with Cernunnos

```
<properties location="classpath:/properties/rdbm.properties">
  <sql-datasource driver="${sqlDriver}" url="${sqlUrl}"
    username="${sqlUsername}" password="${sqlPassword}">
    <sql-query>
      <sql>SELECT * FROM up_user</sql>
      <subtasks><!-- do something useful --></subtasks>
    </sql-query>
  </sql-datasource>
</properties>
```

Cernunnos Is Like LEGOs


- With Cernunnos, you don't have to make it possible for components to combine; you only have to tell them to do so
- When you introduce a new component to Cernunnos, it's instantly capable of combining with every other component -- past, present, and future

Cernunnos Manual

Cernunnos Manual - Mozilla Firefox

file:///C:/HOME/cernunnos-cmd-tool/docs/grammardoc/index.html

grammardoc mycu mycu-backdoor uPortal webtest-wills webtest-wills/backdoor.... YaleInfo



Cernunnos

Table of Contents

[Introduction](#)
[Request Attributes](#)

Phrases

- [attributeNode](#)
- [commentNode](#)
- [const](#)
- [contentsOf](#)
- [cm](#)
- [doc](#)
- [false](#)
- [file](#)
- [grammar](#)
- [groovy](#)
- [isNull](#)
- [jexl](#)
- [indi](#)
- [newDoc](#)
- [newInstance](#)
- [not](#)
- [null](#)
- [parseInt](#)
- [req](#)
- [seq](#)
- [singleNode](#)
- [sql](#)
- [sysprop](#)
- [textNode](#)
- [url](#)
- [valueOf](#)

Cernunnos Manual

<file-iterator>

Description:

Builds a collection of file names within a specified directory and iterates over them. Use optional pattern expressions to specify groups of files to include or exclude.

Reagents:

Name	XPath	Description	Reagent Type	Expected Type	Required
ATTRIBUTE_NAME	@attribute-name	Each file will be registered as a request attribute under this name during iteration.	PHRASE	java.lang.String	No
DIR	@dir	File system location of a directory from which to begin matching files. The default is the directory from which Java is executing.	PHRASE	java.lang.String	No
INCLUDES	@includes	Optional comma-separated list of pattern expressions specifying files to include (e.g. *, *.jpg, **/*.java). Files that match the pattern must be included, even if they also match the EXCLUDES pattern. The default is ** (i.e. all files).	PHRASE	java.lang.String	No
EXCLUDES	@excludes	Optional comma-separated list of pattern expressions specifying files to exclude from the result set (e.g. *, *.jpg, **/*.java). No files will be excluded if this phrase is omitted.	PHRASE	java.lang.String	No
SUBTASKS	*	The set of tasks that are children of this task.	NODE_LIST	java.util.List	No

Examples:

Copies all files from 'some_folder' to 'another_folder':

```
<file-iterator dir="some_folder">
  <copy-file to-dir="another_folder"/>
</file-iterator>
```

Find: Highlight all Match case

Done

Do One Thing Well

- Cernunnos aims to make software less costly by taking on Implied Codemass
- Leave everything else to others
- Put good code in contact with other good code, get out of the way

Other Good Code



Spring Framework



<dom4j>

JavaMail

Spring LDAP

NekoHTML

commons
DBCP

commons
logging

Questions?



Drew Wills

drew@unicon.net

<http://cernunnos.googlecode.com>