.NET @ apache.org

ApacheCon NA - Vancouver BC
November 2011 2:30p

husted@apache.org
http://www.**slideshare**.net/**ted.husted**

.NET @ apache.org

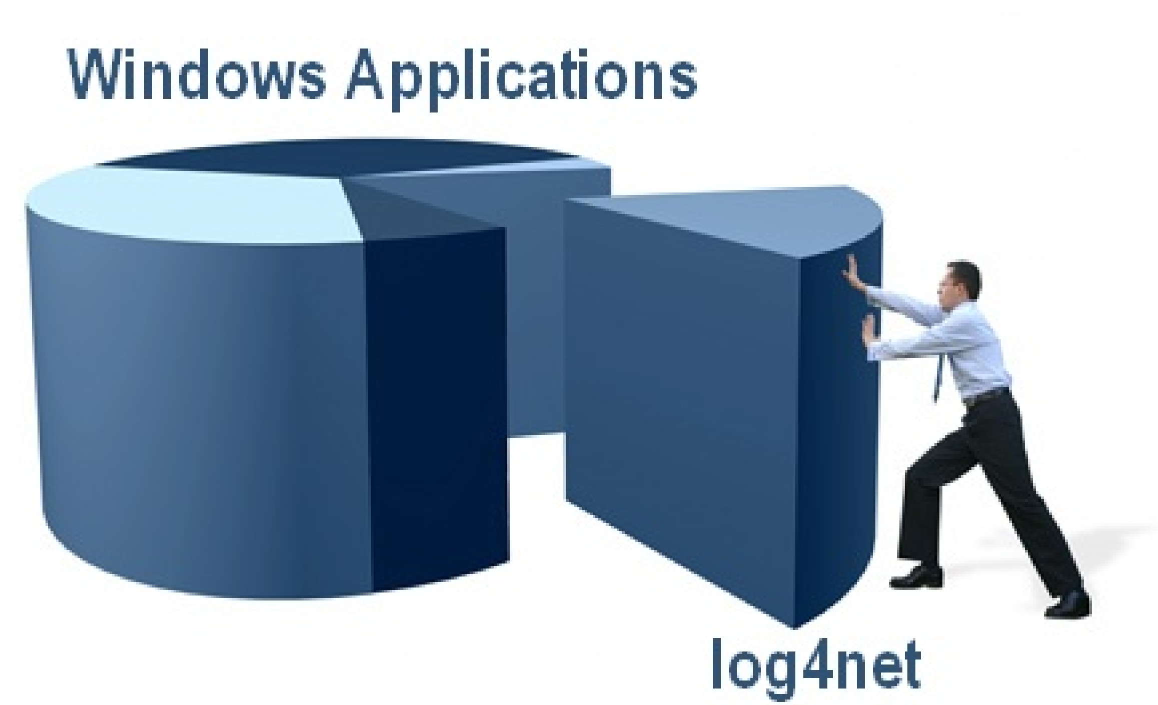
Like it or not, many open source developers are moving to the Microsoft .NET platform, and we're bringing our favorite tools with us!

In this session, we look inside ASF projects that are creating software for .NET and Mono, like Logging and Lucene.net -- to show you how to create leading-edge ASP.NET applications with ASF open source libraries, and how you can integrate with other appications using Thrift, Chemistry/DotCMIS, QPid or ActiveMQ.

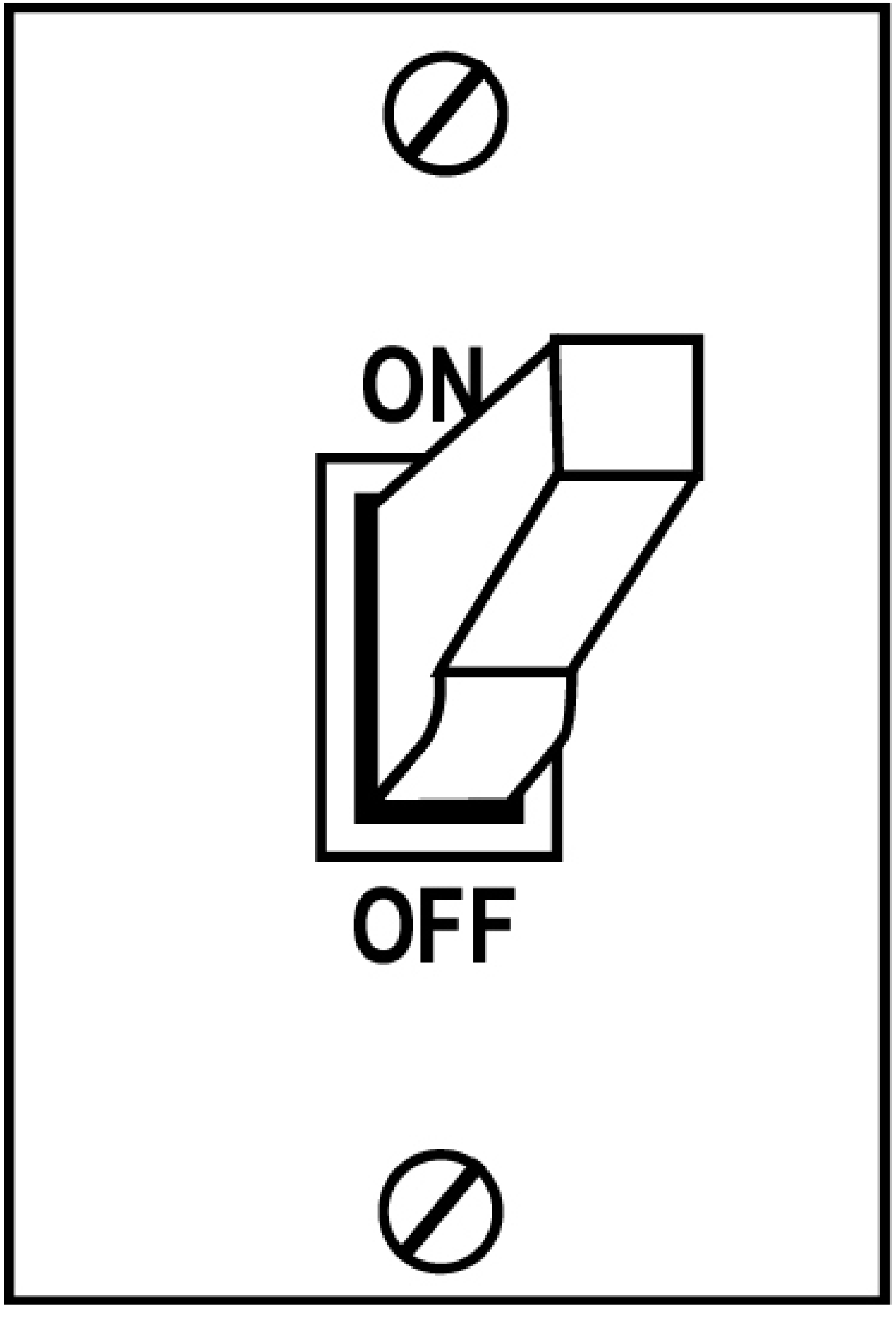
We'll also look at integrating other .NET open source projects, like Spring.NET, NVelocity, and JayRock, into your C# application to create a complete open source .NET stack.

http://logging.apache.org

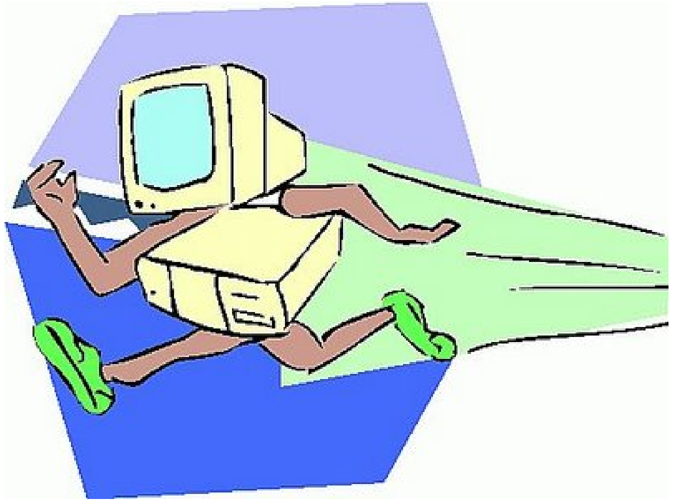
Logging Services



http://balanagaraj.wordpress.com/


ON
OFF

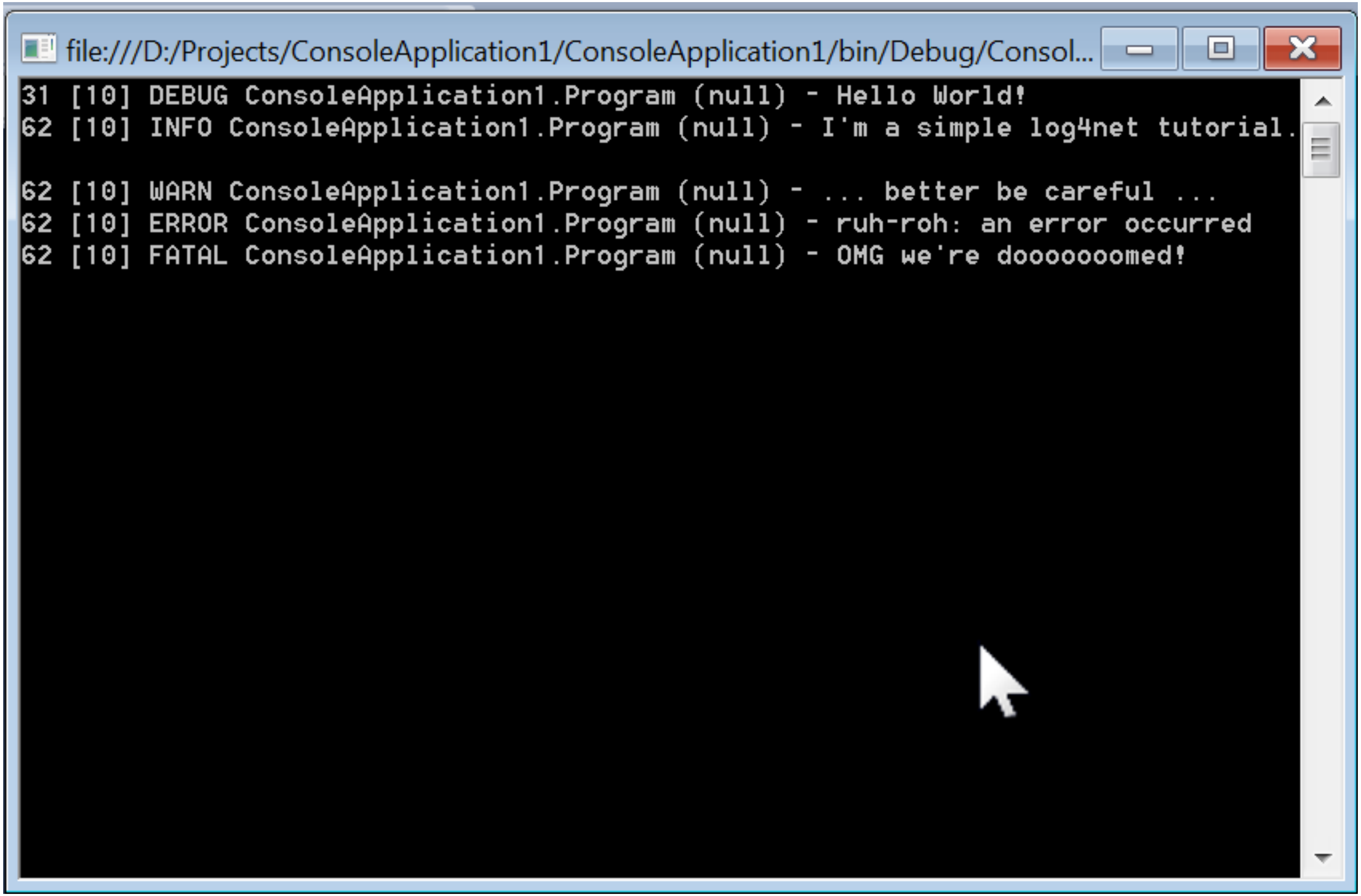
404
404



Hello World - Log4Net

```
using System;
namespace log4net_console
{
 class Program
 {
   static void Main( string[] args )
   {
    log4net.Config.BasicConfigurator.Configure();
    log4net.ILog log = log4net.LogManager.GetLogger (typeof
(Program));
    log.Debug( "Hello World!" );
    log.Info( "I'm a simple log4net tutorial." );
    log.Warn( "... better be careful ..." );
    log.Error( "ruh-roh: an error occurred" );
    log.Fatal( "OMG we're dooooooomed!" );
    Console.ReadLine();  // so you can read the output
   }
 }
}
```

http://www.beefycode.com/post/Log4Net-Tutorial-pt-1-Getting-Started.aspx



file:///D:/Projects/ConsoleApplication1/ConsoleApplication1/bin/Debug/Consol...

```
31 [10] DEBUG ConsoleApplication1.Program (null) - Hello World!
62 [10] INFO ConsoleApplication1.Program (null) - I'm a simple log4net tutorial.

62 [10] WARN ConsoleApplication1.Program (null) - ... better be careful ...
62 [10] ERROR ConsoleApplication1.Program (null) - ruh-roh: an error occurred
62 [10] FATAL ConsoleApplication1.Program (null) - OMG we're dooooooomed!
```

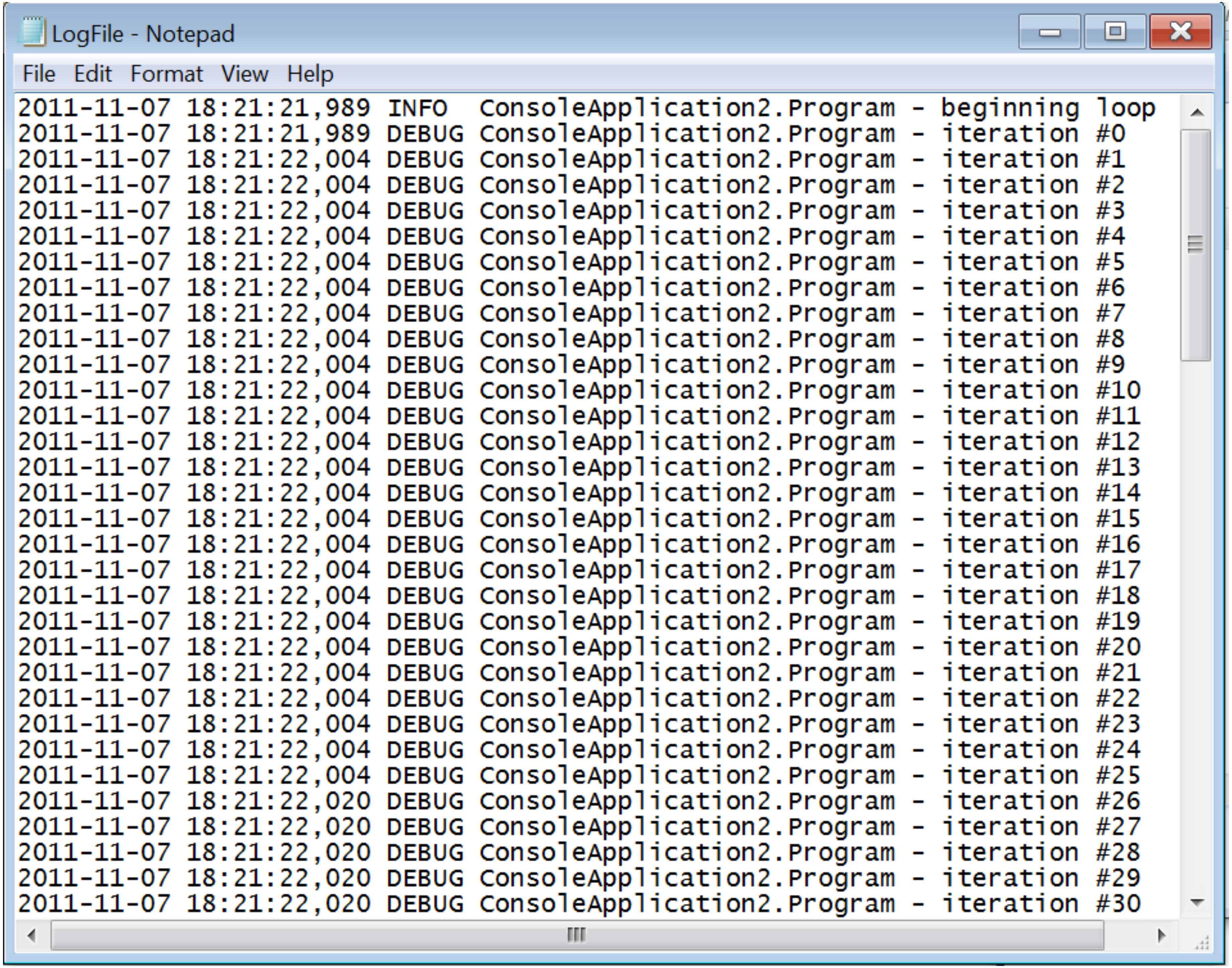
Web.config

```
<appender name="RollingLogFileAppender" type="
log4net.Appender.RollingFileAppender">
  <threshold value="All" />
  <file value="C:\LogFile" />
  <appendToFile value="true" />
  <rollingStyle value="Date" />
  <datePattern value="'.'yyyyMMdd'.log.txt'" />
  <lockingModel type="log4net.Appender.
FileAppender+MinimalLock" />
  <layout type="log4net.Layout.PatternLayout">
  <conversionPattern
value="%date %-5level %logger - %message%newline" />
  </layout>
</appender>
```

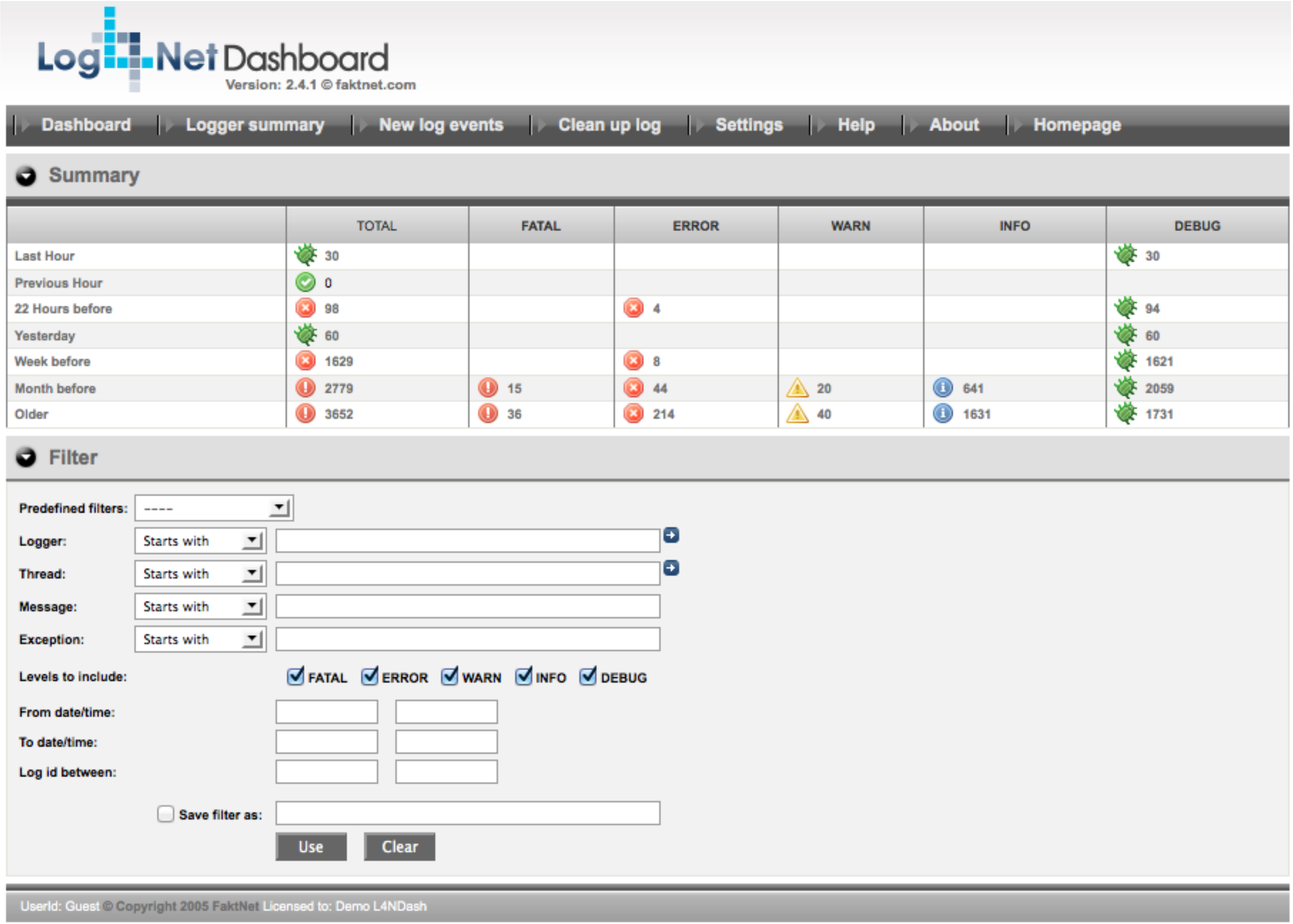
Program.cs

```
using System;
namespace Tutorial2_BasicXmlConfiguration
{
 class Program
 {
  static void Main( string[] args )
  {
   log4net.Config.XmlConfigurator.Configure();
   log4net.ILog log = log4net.LogManager.GetLogger (typeof(Program));
   log.Info( "beginning loop" );
   for( int c = 0; c < 100; ++c )
   {
    log.DebugFormat( "iteration #{0}", c );
   }
    log.Info( "loop has completed" );
    Console.ReadLine();
   }
 }
}
```

// http://www.beefycode.com/post/Log4Net-Tutorial-pt-2-Basic-XML-Configuration.aspx



```
2011-11-07 18:21:21,989 INFO  ConsoleApplication2.Program - beginning loop
2011-11-07 18:21:21,989 DEBUG ConsoleApplication2.Program - iteration #0
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #1
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #2
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #3
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #4
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #5
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #6
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #7
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #8
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #9
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #10
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #11
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #12
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #13
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #14
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #15
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #16
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #17
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #18
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #19
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #20
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #21
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #22
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #23
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #24
2011-11-07 18:21:22,004 DEBUG ConsoleApplication2.Program - iteration #25
2011-11-07 18:21:22,020 DEBUG ConsoleApplication2.Program - iteration #26
2011-11-07 18:21:22,020 DEBUG ConsoleApplication2.Program - iteration #27
2011-11-07 18:21:22,020 DEBUG ConsoleApplication2.Program - iteration #28
2011-11-07 18:21:22,020 DEBUG ConsoleApplication2.Program - iteration #29
2011-11-07 18:21:22,020 DEBUG ConsoleApplication2.Program - iteration #30
```



Summary

	TOTAL	FATAL	ERROR	WARN	INFO	DEBUG
Last Hour	30					30
Previous Hour	0					
22 Hours before	98		4			94
Yesterday	60					60
Week before	1629		8			1621
Month before	2779	15	44	20	641	2059
Older	3652	36	214	40	1631	1731

Filter

Predefined filters: ----

Logger: Starts with

Thread: Starts with

Message: Starts with

Exception: Starts with

Levels to include: ☑ FATAL ☑ ERROR ☑ WARN ☑ INFO ☑ DEBUG

From date/time:

To date/time:

Log id between:

□ Save filter as:

UserId: Guest © Copyright 2005 FaktNet Licensed to: Demo L4NDash

http://www.l4ndash.com/

Installation

ZIP Archive

- **Binary**
  - **\bin**
  - **\doc**
- **Source**
  - **\examples**
  - **\src**
  - **\tests**

NuGet

- Log4Net
- Log4Net Rolling Appender

Installation

ZIP Archive

- **Binary**
  - **\bin**
  - **\doc**
- **Source**
  - **\examples**
  - **\src**
  - **\tests**

NuGet

- Log4Net
- Log4Net Rolling Appender
- Twitter Appender

Log4net

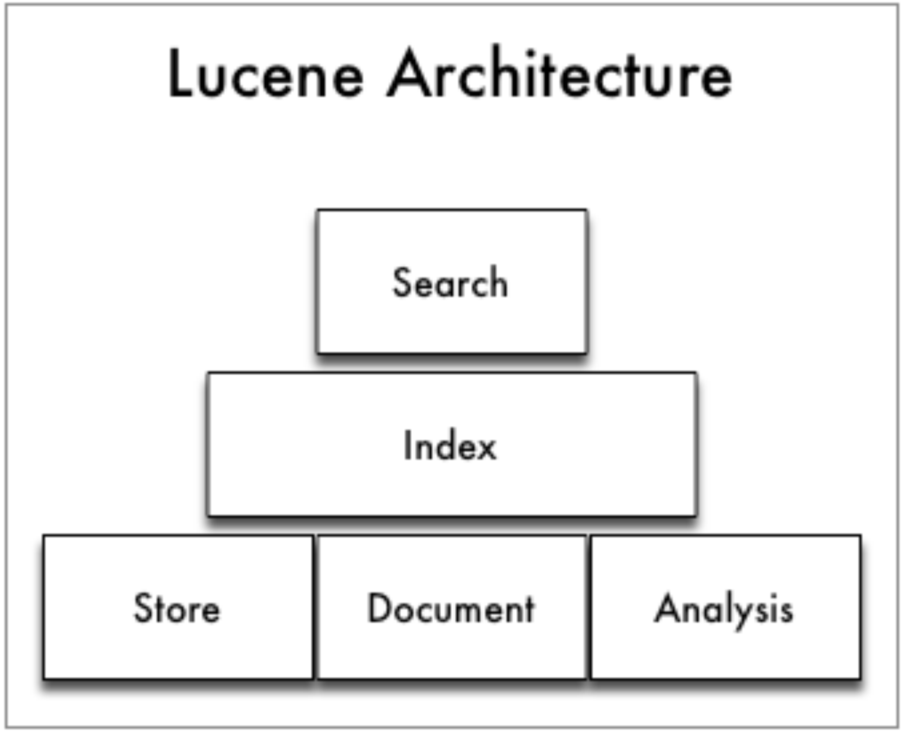
- When to use it
  - You need to **analyze** the function or performance of an application
- What it does
  - Writes **statements** to a file either when certain lines are processed or when errors occur
- How it works
  - Log4net uses a flexible array of **providers** to capture and categorize logging statements
- Where does it fit in
  - Logging is a **crosscutting** concern that can occur at any application layer

Log4Net Resources

- Beefy Code Log4Net Tutorial (7 parts)
  - http://www.beefycode.com/post/Log4Net-Tutorial-pt-1-Getting-Started.aspx
- The Code Project Log4Net Tutorial
  - http://www.codeproject.com/KB/dotnet/Log4net_Tutorial.aspx
- Log4j Logging Mechanism (Kunal Dabir)
  - http://www.slideshare.net/kunal.dabir/log4j-logging-mechanism?src=related_normal&rel=1847004

LUCENE.net™

SEARCH ENGINE LIBRARY



Lucene Architecture

Search

Index

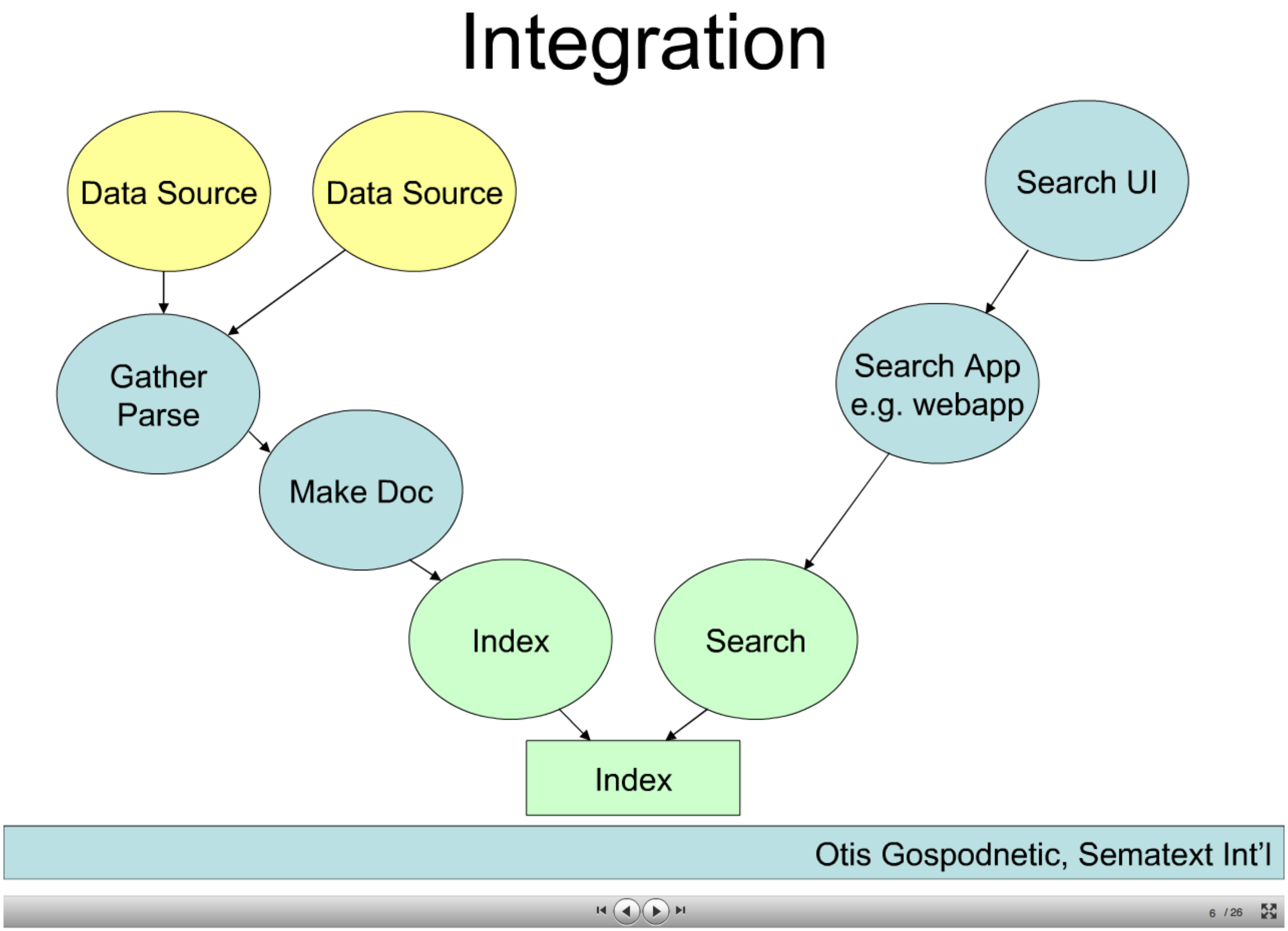
Store

Document

Analysis

http://lemonodor.com/archives/001361.html

SEARCH
DOCUMENTS



Integration

Data Source

Data Source

Gather
Parse

Make Doc

Index

Search UI

Search App
e.g. webapp

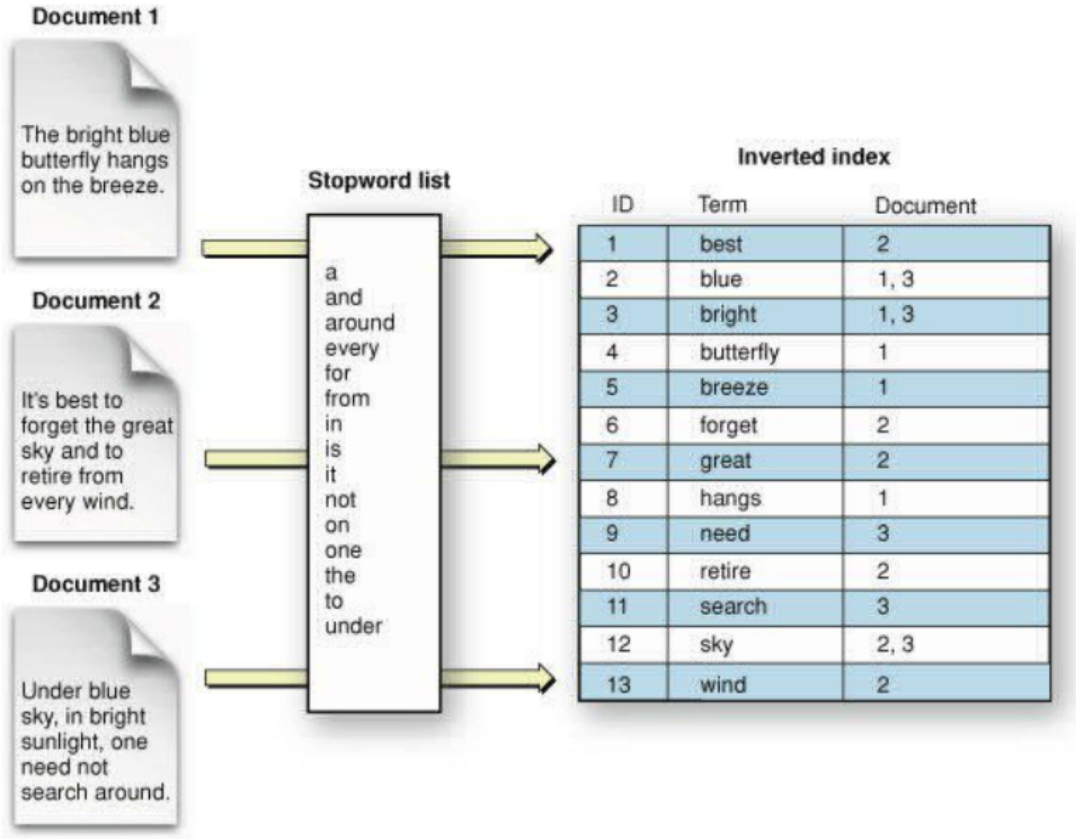
Search

Index

Otis Gospodnetic, Sematext Int’l

http://www.slideshare.net/otisg/lucene-introduction

Inverted Index



Document 1

The bright blue butterfly hangs on the breeze.

Document 2

It's best to forget the great sky and to retire from every wind.

Document 3

Under blue sky, in bright sunlight, one need not search around.

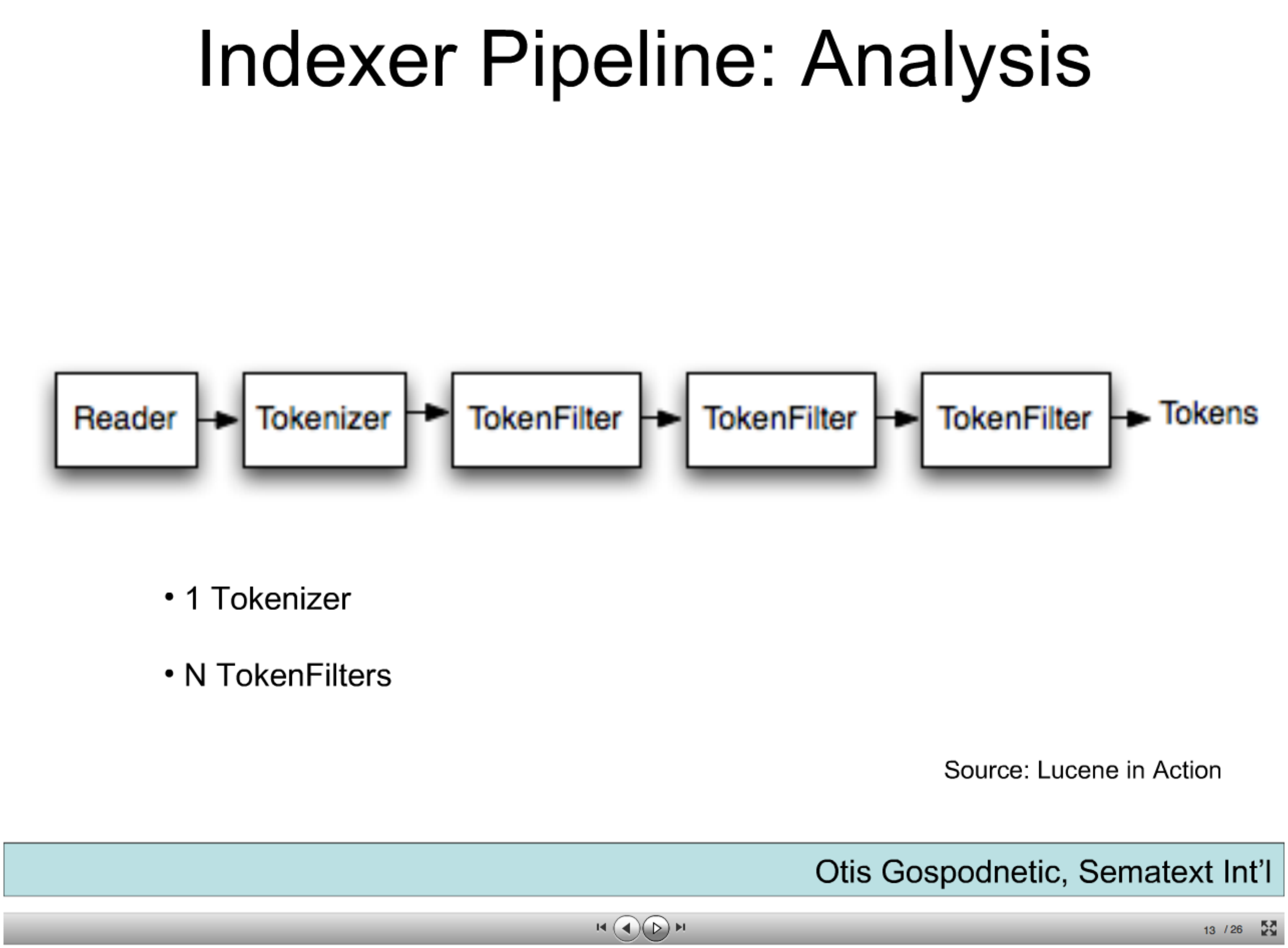
Stopword list

a
and
around
every
for
from
in
is
it
not
on
one
the
to
under

Inverted index

ID	Term	Document
1	best	2
2	blue	1, 3
3	bright	1, 3
4	butterfly	1
5	breeze	1
6	forget	2
7	great	2
8	hangs	1
9	need	3
10	retire	2
11	search	3
12	sky	2, 3
13	wind	2

Source: developer.apple.com



Indexer Pipeline: Analysis

Reader → Tokenizer → TokenFilter → TokenFilter → TokenFilter → Tokens

- 1 Tokenizer
- N TokenFilters

Source: Lucene in Action

Otis Gospodnetic, Sematext Int’l

http://www.slideshare.net/otisg/lucene-introduction

Documents and Fields

```
// Documents and Fields
var fordFiesta = new Document();
fordFiesta.Add(new Field("Id", "1", Field.Store.YES, Field.Index.NOT_ANALYZED));
fordFiesta.Add(new Field("Make", "Ford", Field.Store.YES, Field.Index.ANALYZED));
fordFiesta.Add(new Field("Model", "Fiesta", Field.Store.YES, Field.Index.ANALYZED));
var fordFocus = new Document();
fordFocus.Add(new Field("Id", "2", Field.Store.YES, Field.Index.NOT_ANALYZED));
fordFocus.Add(new Field("Make", "Ford", Field.Store.YES, Field.Index.ANALYZED));
fordFocus.Add(new Field("Model", "Focus", Field.Store.YES, Field.Index.ANALYZED));

var vauxhallAstra = new Document();
vauxhallAstra.Add(new Field("Id", "3", Field.Store.YES, Field.Index.NOT_ANALYZED));
vauxhallAstra.Add(new Field("Make", "Vauxhall", Field.Store.YES, Field.Index.ANALYZED));
vauxhallAstra.Add(new Field("Model", "Astra", Field.Store.YES, Field.Index.ANALYZED));
```

http://www.d80.co.uk/post/2011/03/29/LuceneNet-Tutorial.aspx

Directories, Analyzers, and Writers

```
// Directory
string directoryLocation = Environment.CurrentDirectory + "\\LuceneIndex";
Directory directory = FSDirectory.Open(new DirectoryInfo(directoryLocation));

// Analyzer
Analyzer analyzer = new StandardAnalyzer(Version.LUCENE_29);

// Writer
var writer = new IndexWriter(directory, analyzer, true, IndexWriter.MaxFieldLength.LIMITED);
writer.AddDocument(fordFiesta);
writer.AddDocument(fordFocus);
writer.AddDocument(vauxhallAstra);
writer.Optimize();
writer.Close();
Console.WriteLine("Index stored at " + directoryLocation);
```

Parse, Query, and Search

```
// Read Index
IndexReader indexReader = IndexReader.Open(directory, true);
// Parse and Query Index
var queryParser = new QueryParser(Version.LUCENE_29, "Make", analyzer);
var query = queryParser.Parse("Ford");
Searcher indexSearch = new IndexSearcher(indexReader);
// Search Index and review Hits
TopDocs resultDocs = indexSearch.Search(query, indexReader.MaxDoc());
var hits = resultDocs.scoreDocs;
foreach (var hit in hits) {
 var documentFromSearcher = indexSearch.Doc(hit.doc);
 Console.WriteLine(documentFromSearcher.Get("Make") + " " +
  documentFromSearcher.Get("Model"));
}
```

Lucene.net

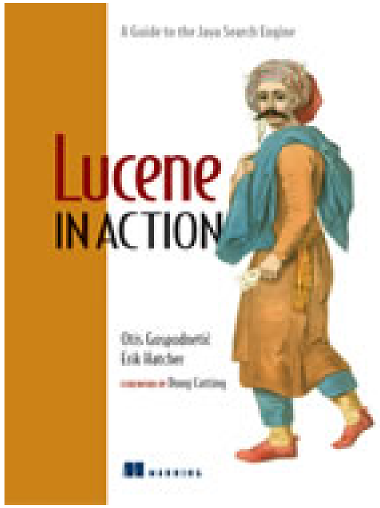
- When to use it
  - You need to **search** data in a flexible and powerful way
- What it does
  - **Indexes** data for retrieval from primary storage
- How it works
  - Lucene "**tokenizes**" strings into an inverted index
- Where does it fit in
  - Data access and Data storage layer
    - Searches are logical queries that access indexes stored as a secondary data source

Lucene.net Resources

- Lucene.net Tutorial (Dan's Dev Blog)
    - http://www.d80.co.uk/post/2011/03/29/LuceneNet-Tutorial.aspx
    - http://www.d80.co.uk/post/2011/09/08/LuceneNet-Video-Tutorial.aspx
- Lucene Introduction
    - http://www.slideshare.net/otisg/lucene-introduction
- Simple Lucene - Lucene.net made easy
    - http://blogs.planetcloud.co.uk/mygreatdiscovery/post/SimpleLucene-e28093-Lucenenet-made-easy.asp

Lucene.net Resources

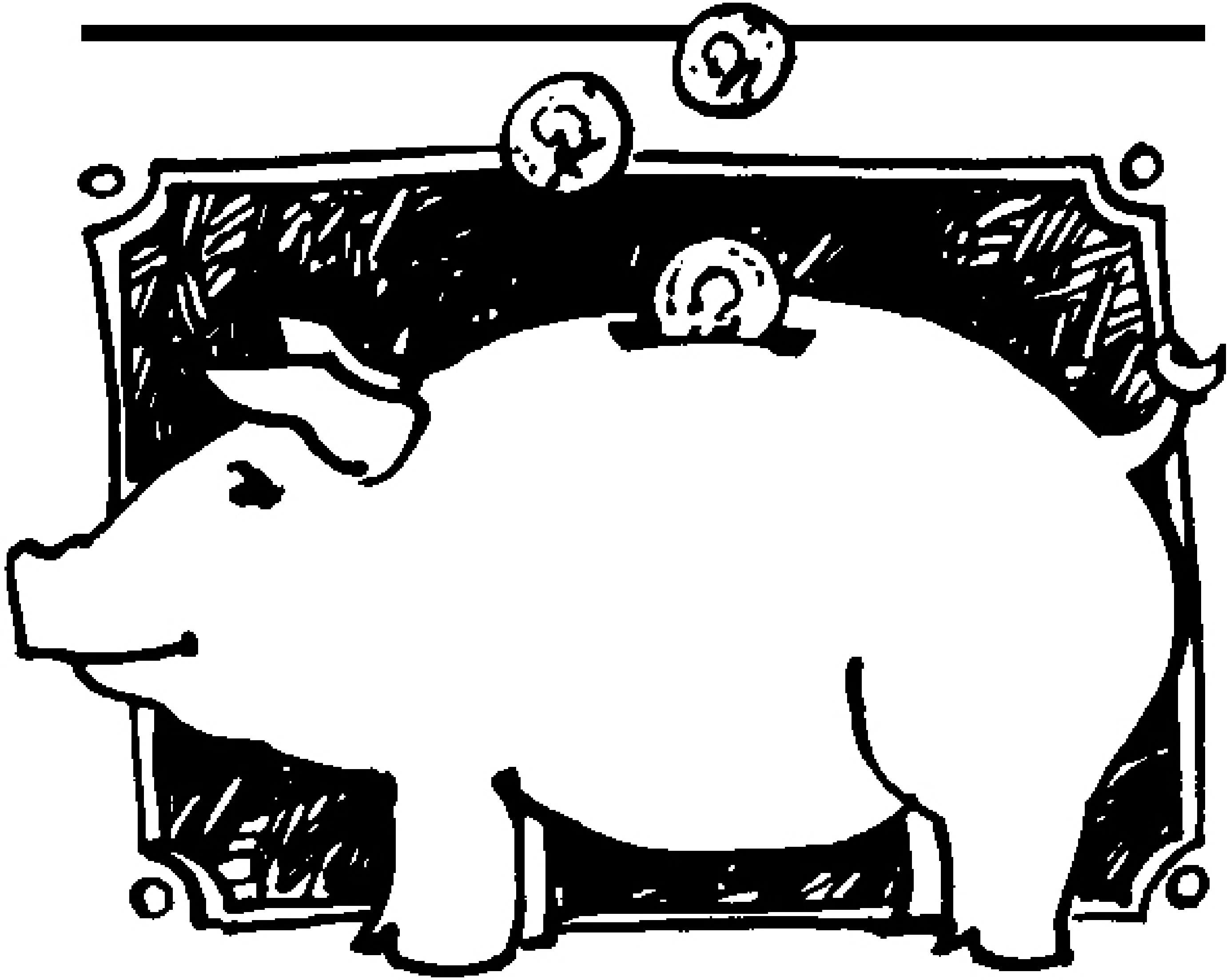
- Lucene.net Tutorial (Dan's Dev Blog)
  - http://www.d80.co.uk/post/2011/03/29/LuceneNet-Tutorial.aspx
  - http://www.d80.co.uk/post/2011/09/08/LuceneNet-Video-Tutorial.aspx

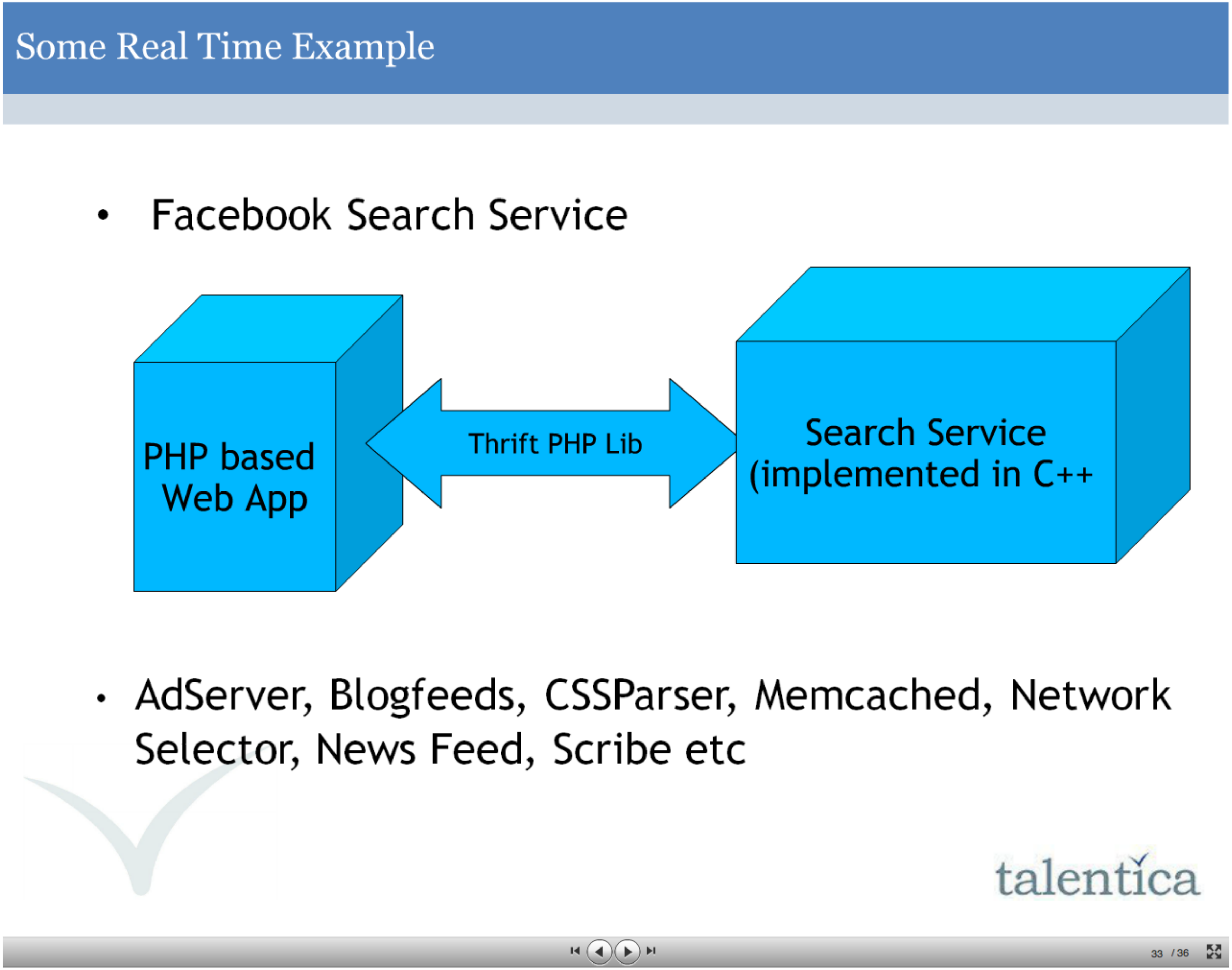


- Lucene Introduction
  - http://www.slideshare.net/otisg/lucene-introduction

- Simple Lucene - Lucene.net made easy
  - http://blogs.planetcloud.co.uk/mygreatdiscovery/post/SimpleLucene-e28093-Lucenenet-made-easy.asp

THRIFT!





Some Real Time Example

- Facebook Search Service

PHP based Web App

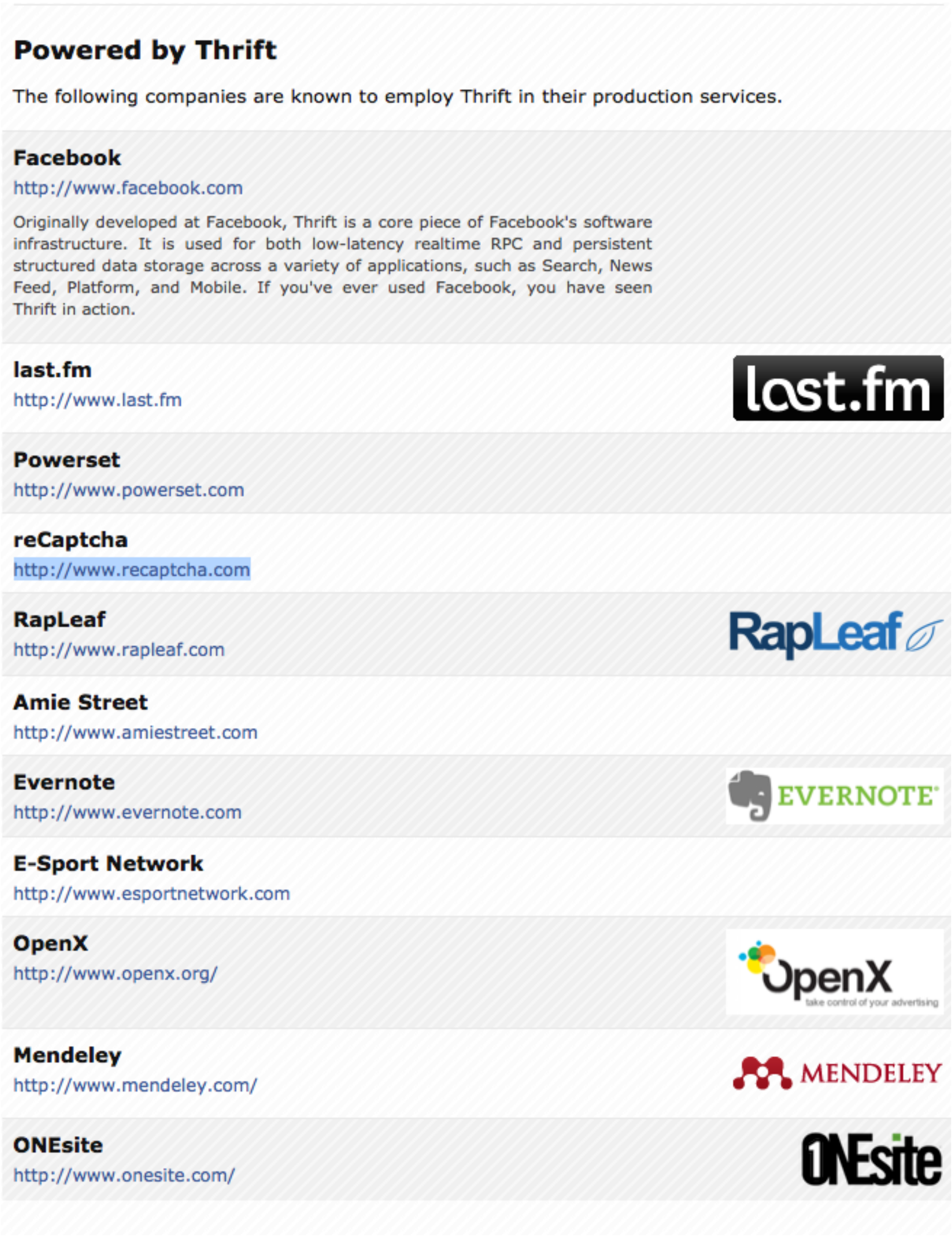
Thrift PHP Lib

Search Service (implemented in C++

- AdServer, Blogfeeds, CSSParser, Memcached, Network Selector, News Feed, Scribe etc

talentica

www.slideshare.net/talentica/building-scalable-and-language-independent-java-services-using-apache-thrift-6286467



Powered by Thrift

The following companies are known to employ Thrift in their production services.

Facebook
http://www.facebook.com

Originally developed at Facebook, Thrift is a core piece of Facebook's software infrastructure. It is used for both low-latency realtime RPC and persistent structured data storage across a variety of applications, such as Search, News Feed, Platform, and Mobile. If you've ever used Facebook, you have seen Thrift in action.

last.fm
http://www.last.fm

Powerset
http://www.powerset.com

reCaptcha
http://www.recaptcha.com

RapLeaf
http://www.rapleaf.com

Amie Street
http://www.amiestreet.com

Evernote
http://www.evernote.com

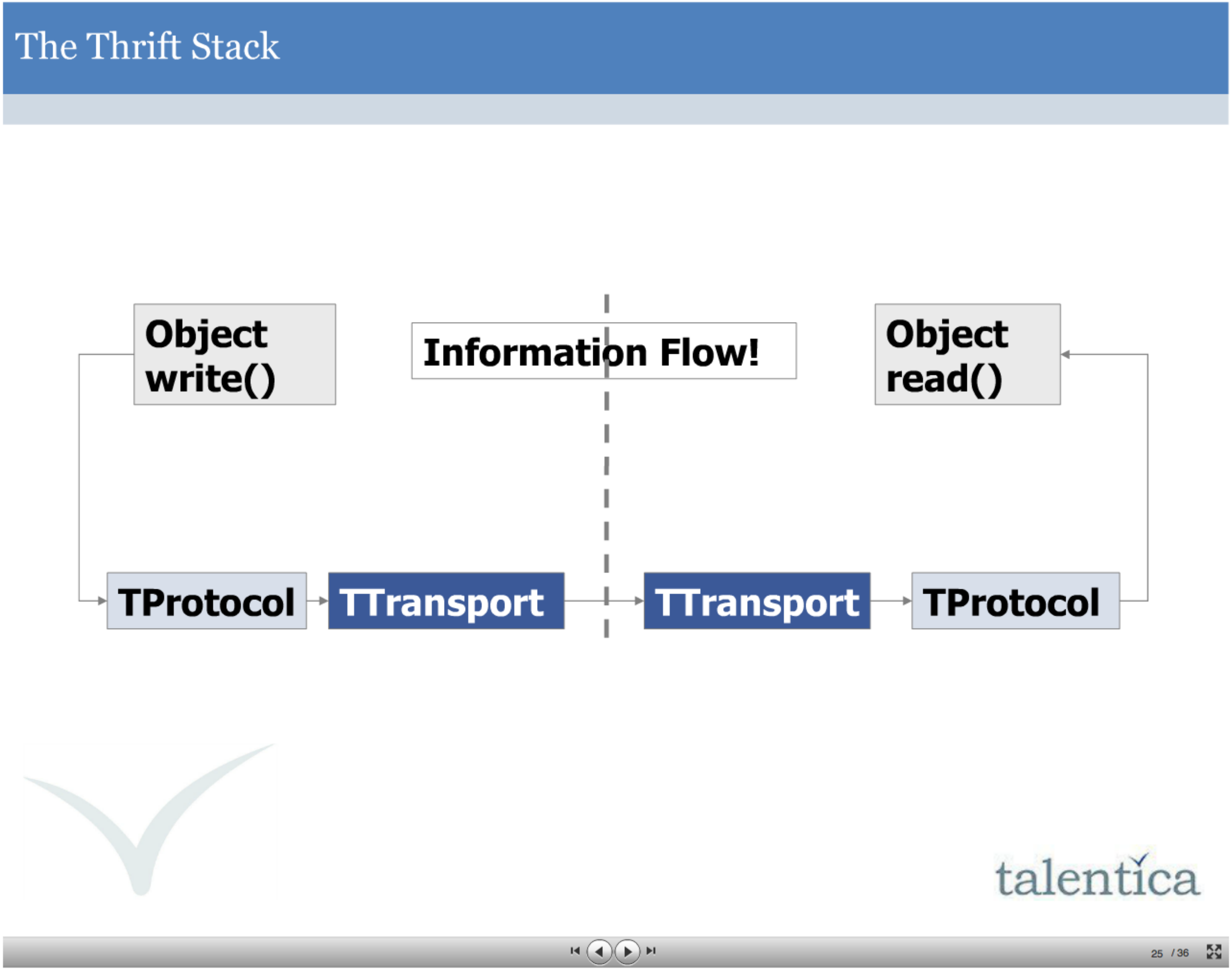
E-Sport Network
http://www.esportnetwork.com

OpenX
http://www.openx.org/

Mendeley
http://www.mendeley.com/

ONEsite
http://www.onesite.com/

http://thrift.apache.org/about/



The Thrift Stack

Object write()

Information Flow!

Object read()

TProtocol

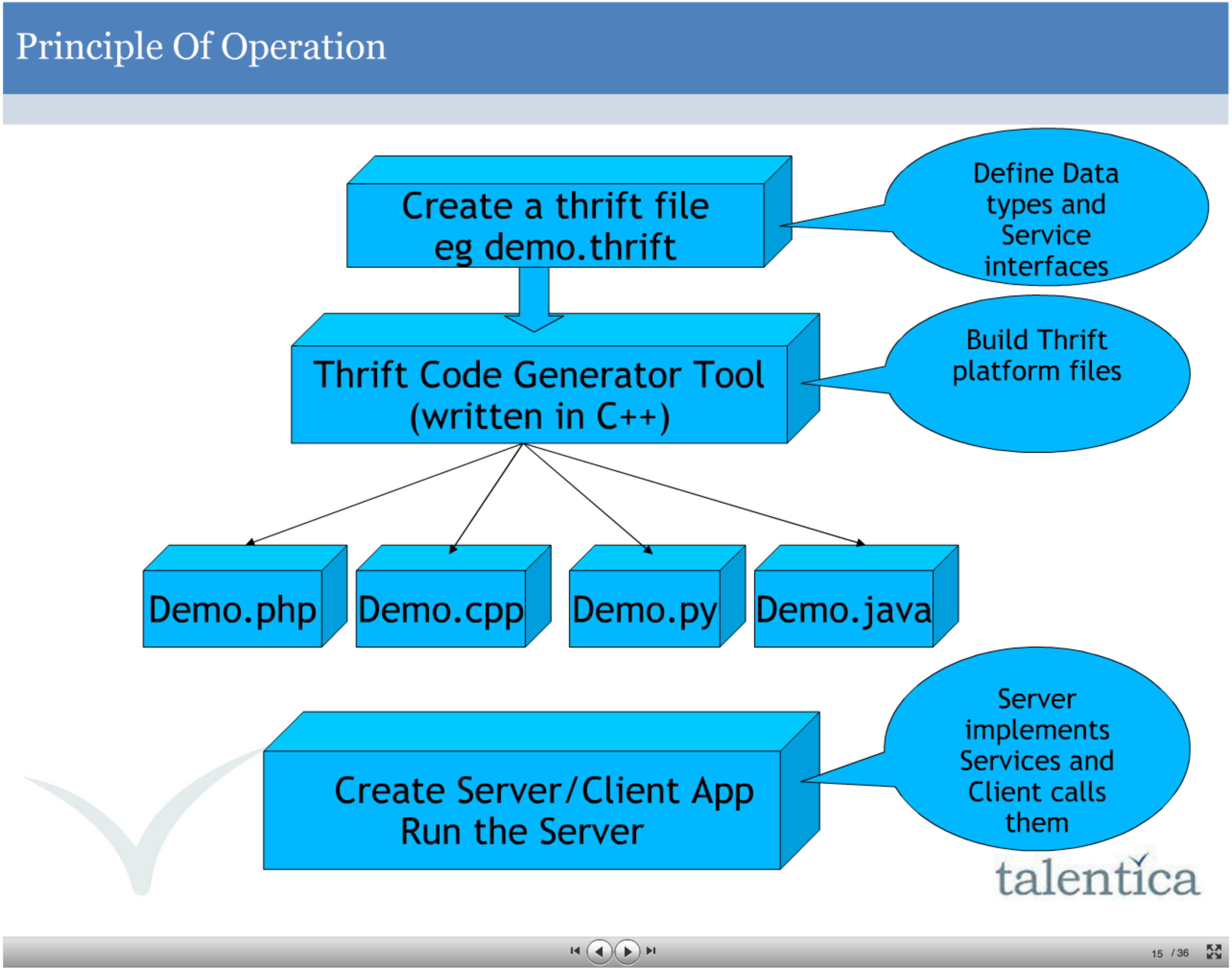
TTransport

TTransport

TProtocol

25 / 36

www.slideshare.net/talentica/building-scalable-and-language-independent-java-services-using-apache-thrift-6286467



Principle Of Operation

Create a thrift file
eg demo.thrift

Define Data types and Service interfaces

Thrift Code Generator Tool
(written in C++)

Build Thrift platform files

Demo.php

Demo.cpp

Demo.py

Demo.java

Create Server/Client App
Run the Server

Server implements Services and Client calls them

talentica

15 / 36

www.slideshare.net/talentica/building-scalable-and-language-independent-java-services-using-apache-thrift-6286467

time.thrift

```
# time.thrift namespace java tserver.gen
typedef i64 Timestamp
service TimeServer {
  Timestamp time()
}
```

http://nbonvin.wordpress.com/2009/03/08/simple-thrift-tutorial/

Generate the code

```
thrift --gen java time.thrift
```

TimeServerImpl.java

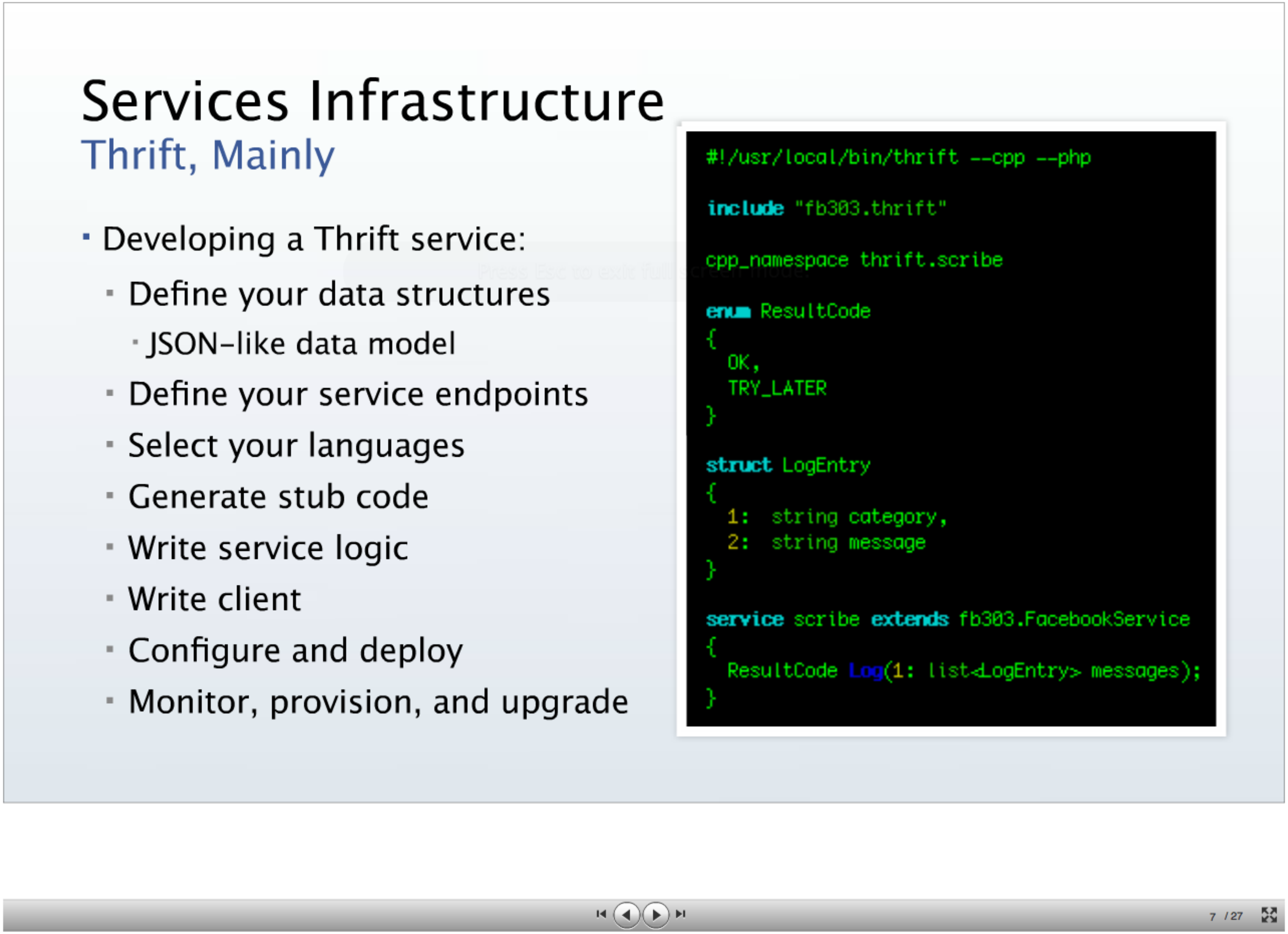
```
package server;
import org.apache.thrift.TException;
import tserver.gen.*;

class TimeServerImpl implements  TimeServer.Iface {
  @Override
  public long time() throws TException {
    long time = System.currentTimeMillis();
    System.out.println("time() called: " + time);
    return time;
  }
}
```

Server.java

```
package server;
import java.io.IOException;
import org.apache.thrift.protocol.TBinaryProtocol;
import org.apache.thrift.protocol.TBinaryProtocol.Factory;
import org.apache.thrift.server.TServer; import org.apache.thrift.server.TThreadPoolServer;
import org.apache.thrift.transport.TServerSocket;  import org.apache.thrift.transport.TTransportException;
import tserver.gen.TimeServer;
public class Server {
 private void start() {
  try {
   TServerSocket serverTransport = new TServerSocket(7911);
   TimeServer.Processor processor = new TimeServer.Processor(new TimeServerImpl());
   Factory protFactory = new TBinaryProtocol.Factory(true, true);
   TServer server = new TThreadPoolServer(processor, serverTransport, protFactory);
   System.out.println("Starting server on port 7911 ...");
   server.serve();
   } catch (TTransportException e) { e.printStackTrace();
   } catch (IOException e) { e.printStackTrace();
  }
}
public static void main(String args[]) {
  Server srv = new Server(); srv.start(); }}
```

TimeClient.java

```
package client;
import java.net.SocketException;
import org.apache.thrift.TException;
import org.apache.thrift.protocol.TBinaryProtocol;
import org.apache.thrift.protocol.TProtocol; import org.apache.thrift.transport.TSocket;
import org.apache.thrift.transport.TTransport;  import org.apache.thrift.transport.TTransportException;
import tserver.gen.TimeServer.Client;
public class TimeClient {
  private void start() {
   TTransport transport;
   try {
    transport = new TSocket("localhost", 7911);
    TProtocol protocol = new TBinaryProtocol(transport);
     Client client = new Client(protocol);
     transport.open();
     long time = client.time();
    System.out.println("Time from server:" + time);
    transport.close();
   } catch (SocketException e) { e.printStackTrace();  } catch (TTransportException e) {
     e.printStackTrace();  } catch (TException e) {  e.printStackTrace();
   }
}
public static void main(String[] args) {
  TimeClient c = new TimeClient(); c.start();  } }
```



Services Infrastructure

Thrift, Mainly

- Developing a Thrift service:
  - Define your data structures
    - JSON-like data model
  - Define your service endpoints
  - Select your languages
  - Generate stub code
  - Write service logic
  - Write client
  - Configure and deploy
  - Monitor, provision, and upgrade

```
#!/usr/local/bin/thrift --cpp --php

include "fb303.thrift"

cpp_namespace thrift.scribe

enum ResultCode
{
  OK,
  TRY_LATER
}

struct LogEntry
{
  1:  string category,
  2:  string message
}

service scribe extends fb303.FacebookService
{
  ResultCode Log(1: list<LogEntry> messages);
}
```

http://www.slideshare.net/jhammerb/20080611accel

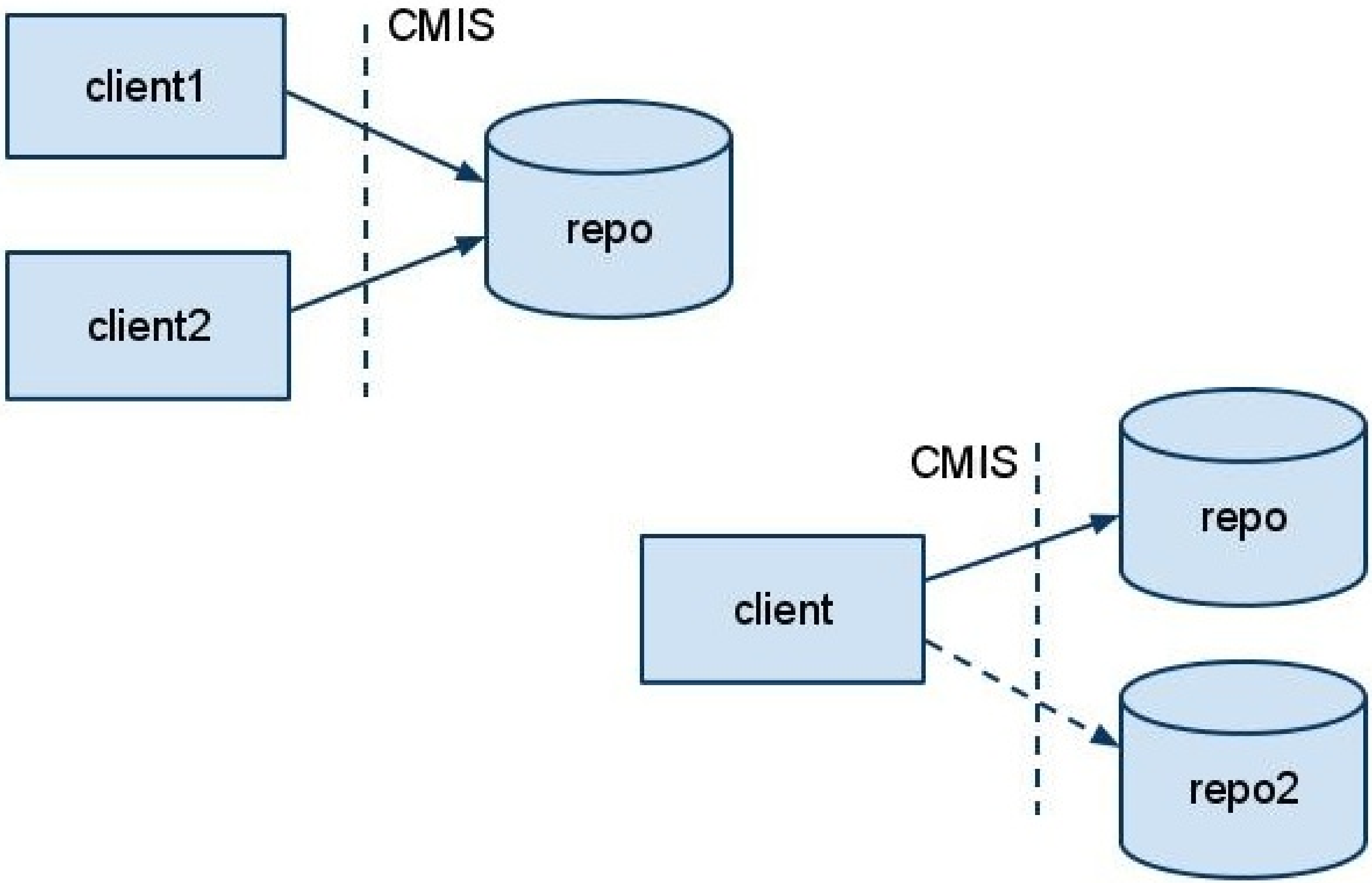
Thrift

- When to use it
  - You need to **invoke an action** between co-located servers, especially on a different language platform.
- What it does
  - **Transfers data** to a procedure on the server using a flexible, lightweight, efficient protocol
- How it works
  - You define the **data transfer object** and service in a JSON-like syntax, and Thrift **generates the code** for client code and stub server code
- Where does it fit in
  - Data source layer
    - Alternative to SOAP, REST, or COM

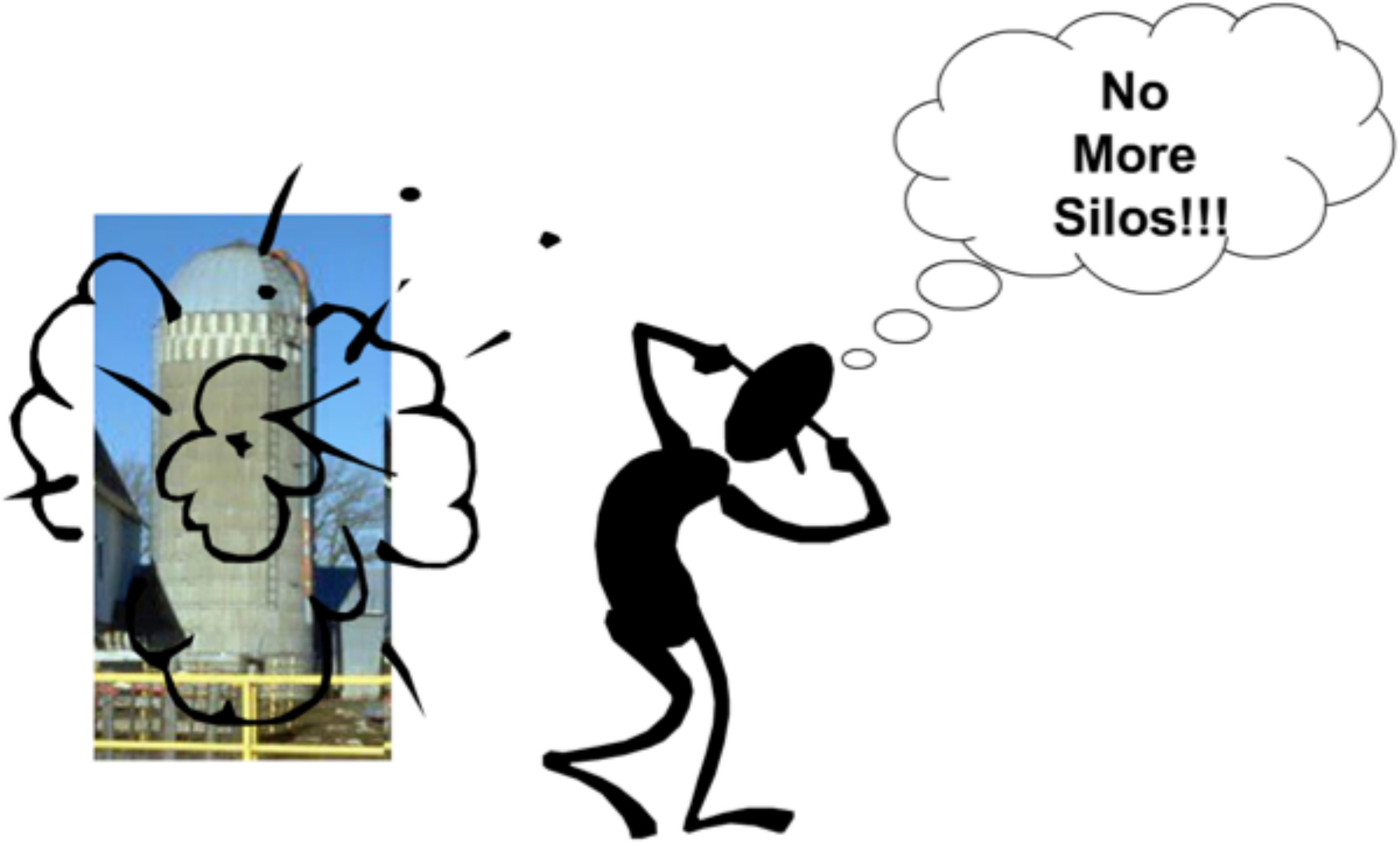
Thrift Resources

- Apache Thrift (Prunicki)
  - http://jnb.ociweb.com/jnb/jnbJun2009.html
- Getting Started with Apache Thrift Tutorial (Rakowski)
  - http://www.thrift.pl/Thrift-tutorial-getting-started.html
- Simple Thrift Tutorial (Bovin)
  - http://nbonvin.wordpress.com/2009/03/08/simple-thrift-tutorial/

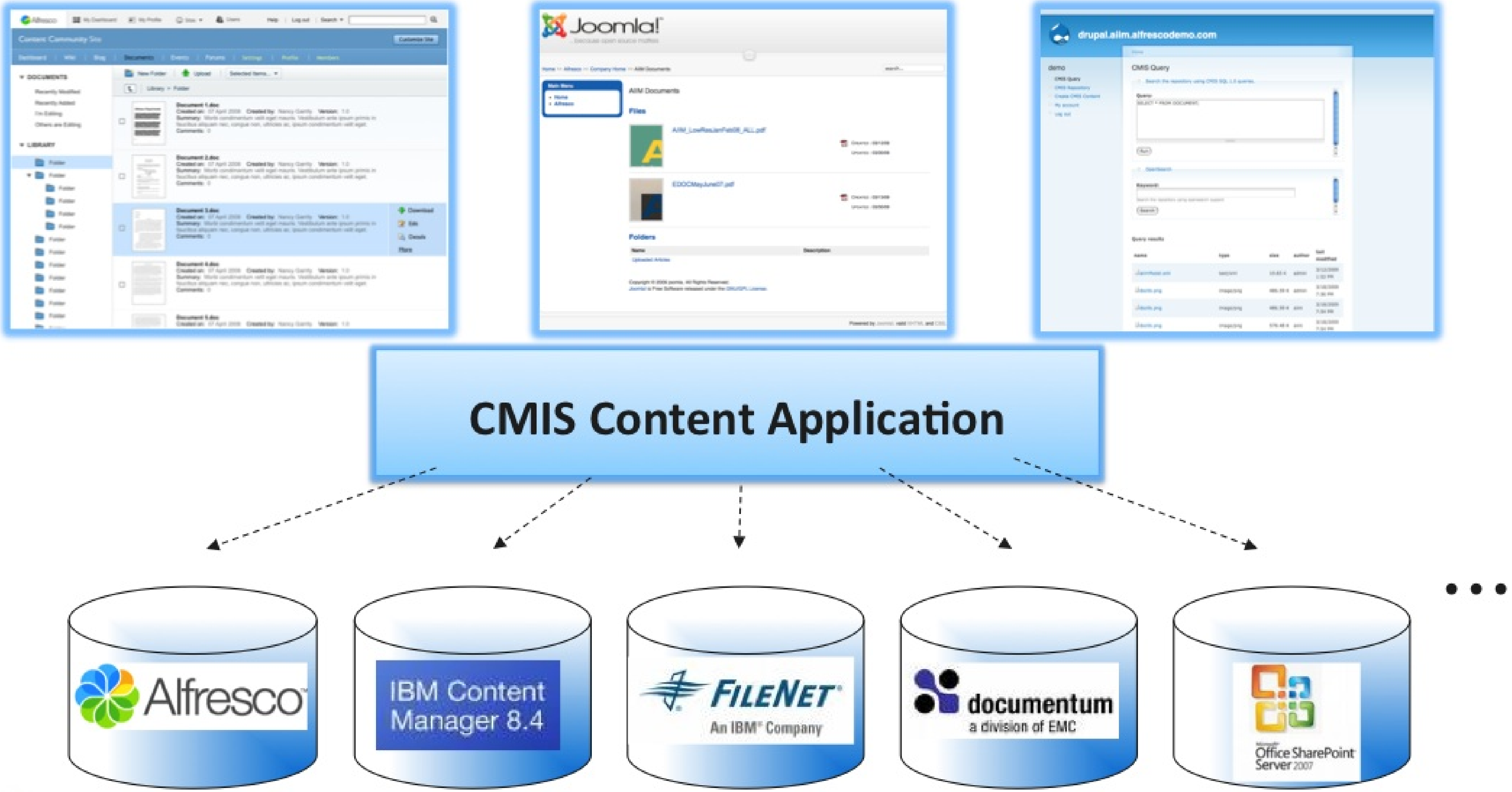
Apache Chemistry™



http://gazarenkov.blogspot.com/



http://www.slideshare.net/pie1120/the-point-of-the-content-interoperability-services-cmis-standard



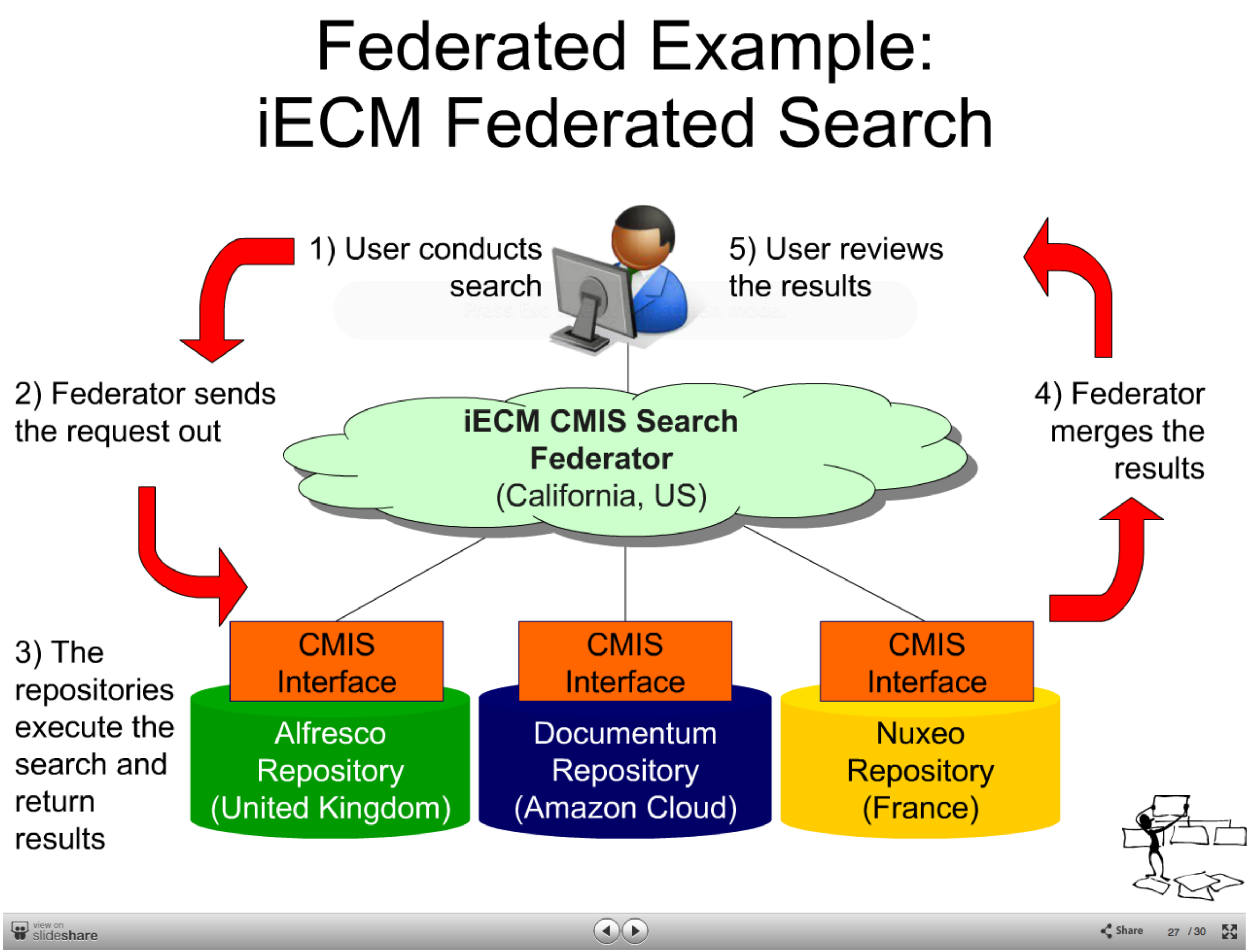
CMIS Content Application

Repository to Repository

Application to Repository

Federated Repositories

http://www.slideshare.net/pie1120/the-point-of-the-content-interoperability-services-cmis-standard



Federated Example: iECM Federated Search

1) User conducts search

2) Federator sends the request out

iECM CMIS Search Federator (California, US)

3) The repositories execute the search and return results

CMIS Interface — Alfresco Repository (United Kingdom)

CMIS Interface — Documentum Repository (Amazon Cloud)

CMIS Interface — Nuxeo Repository (France)

4) Federator merges the results

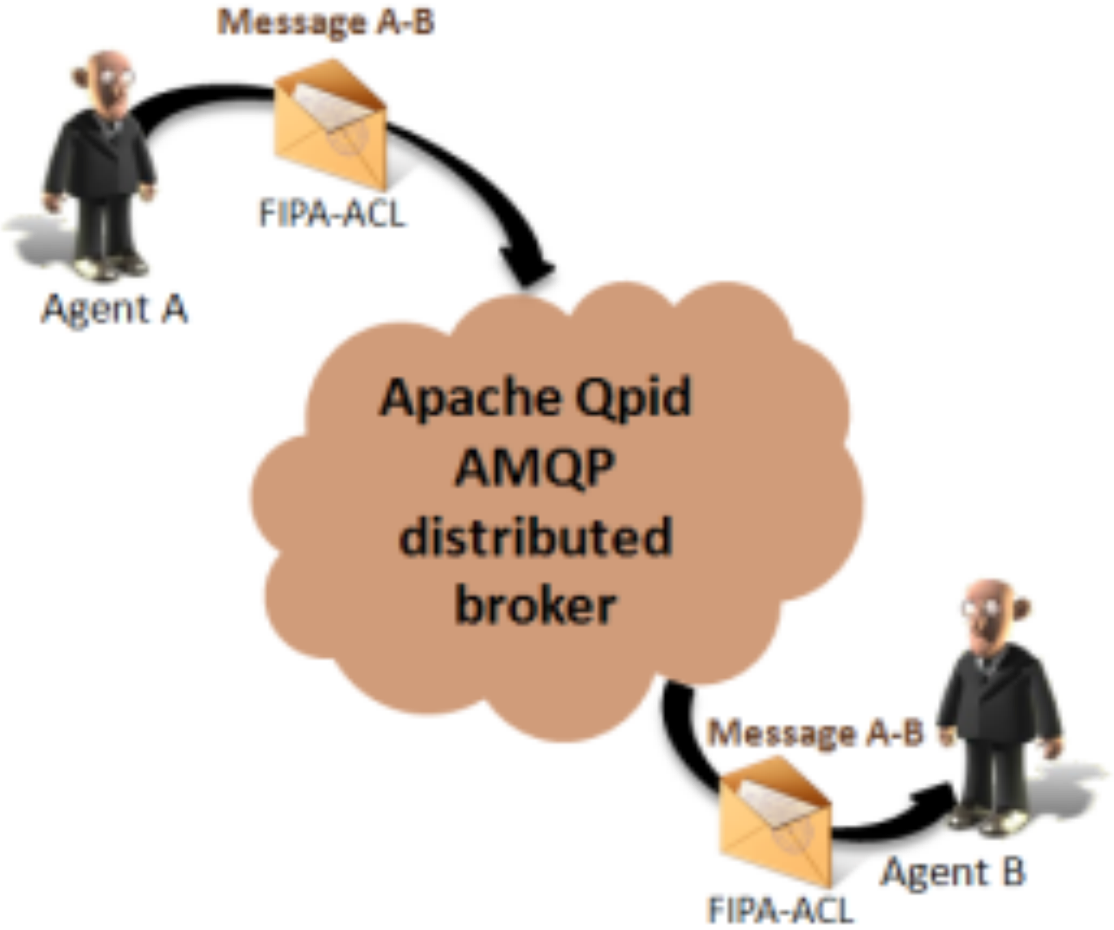
5) User reviews the results

http://www.slideshare.net/pie1120/the-point-of-the-content-interoperability-services-cmis-standard

Chemistry / DotCMIS

- When to use it
    - You need to connect CMIS **repositories**
- What it does
    - DotCMIS implements the **CMIS** client standard
- How it works
    - Exchanges data through **web services or Atom**.
- Where does it fit in
    - Data Access Layer

Qpid



http://gti-ia.upv.es/sma/tools/magentix2/overview.php

Qpid API

- message
-
- connection
-
- session
-
- sender

- receiver

Qpid API

- message - standard fields, custom properties, and content.
-
- connection
-
- session
-
- sender

- receiver

Qpid API

- message - standard fields, custom properties, and content.
-
- connection - network connection to a remote endpoint.
-
- session
-
- sender

- receiver

Qpid API

- message - standard fields, custom properties, and content.
-
- connection - network connection to a remote endpoint.
-
- session - ordered *messages* based on a *connection*.
-
- sender

- receiver

Qpid API

- message - standard fields, custom properties, and content.
-
- connection - network connection to a remote endpoint.
-
- session - ordered *messages* based on a *connection*.
-
- sender *- based on a session for a given target address.*

- receiver

Qpid API

- message - standard fields, custom properties, and content.
-
- connection - network connection to a remote endpoint.
-
- session - ordered *messages* based on a *connection*.
-
- sender *- based on a session for a given target address.*

- receiver *- based on a session for a given source address.*

```
using System;
using Org.Apache.Qpid.Messaging;
namespace Org.Apache.Qpid.Messaging {
class Program {
 static void Main(string[] args) {
 String broker = args.Length > 0 ? args[0] : "localhost:5672";
 String address = args.Length > 1 ? args[1] : "amq.topic";
 Connection connection = null;
 try {
  connection = new Connection(broker);
  connection.Open();
  Session session = connection.CreateSession();
  Receiver receiver = session.CreateReceiver(address);
  Sender sender = session.CreateSender(address);
  sender.Send(new Message("Hello world!"));
  Message message = new Message();
  message = receiver.Fetch(DurationConstants.SECOND * 1);
  Console.WriteLine("{0}", message.GetContent());
  session.Acknowledge(); connection.Close();
  } catch (Exception e) {
    Console.WriteLine("Exception {0}.", e);
    if (connection != null) connection.Close();
}}}}
```

docs.redhat.com/docs/en-US/Red_Hat_Enterprise_MRG/1.3/html-single/Programming_in_Apache_Qpid/index.html

Qpid

- When to use it
  - Various clients need to exchange information with an **incompatible server**.
- What it does
  - Places a sophisticated **broker** between system that can queue and/or transform messages.
- How it works
  - Qpid implements the **AMPQ** industry standard.
- Where does it fit in
  - Data access layer integration

Qpid Resources

- Message Orientated Middleware (Wikipedia)
    - http://en.wikipedia.org/wiki/Message-oriented_middleware
- Advanced Message Queuing Protocol (Wikipedia)
    - http://en.wikipedia.org/wiki/Advanced_Message_Queuing_Protocol
- Programming in Apache Qpid (Red Hat)
- http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_MRG/1.3/html-single/Programming_in_Apache_Qpid/index.html

ActiveMQ C# Client

- When to use it
  - You want to connect a C# client to a **Java Messaging Server** (among others)
- What it does
  - Sends and receives **messages** brokered by a server.
- How it works
  - Exposes a standard **API** and provides connectivity.
- Where does it fit in
  - Data Access Layer integration

- http://activemq.apache.org/

Spring.NET + JayRock + NVelocity

- When to use it
  - You would like an **alternative** to the default ASP.NET framework.
- What it does
  - With Postgres or MongoDB, forms a soup-to-nuts **web application platform**.
- How it works
  - Taken together the platform provides **dependency injection, remote procedure calls, and dynamic, lightweight UI pages.**
- Where does it fit in
  - All levels

Spring.net

```
<objects xmlns="http://www.springframework.net">
<object name="MyMovieLister"
  type="Spring.Examples.MovieFinder.MovieLister,
Spring.Examples.MovieFinder">
</object>
 </objects>
```

```
public static void Main () {
 IApplicationContext ctx = ContextRegistry.GetContext();
 IMovieLister lister = (IMovieLister) ctx.GetObject ("MyMovieLister");
  Movie[] movies = lister.MoviesDirectedBy("Roberto Benigni");
}
```

```
public interface IMovieLister
  IList MoviesDirectedBy (string director);
}
```

http://www.springframework.net/doc-latest/reference/html/quickstarts.html

Jayrock

```
public class HelloWorld : JsonRpcHandler {
[ JsonRpcMethod("greetings") ]
public string Greetings() {
  return "Welcome to Jayrock!";
} }
```

http://jayrock.berlios.de/

Jayrock

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
 <head>
  <title>Hello Jayrock</title>
  <script type="text/javascript" src="json.js"></script>
  <script type="text/javascript" src="helloworld.ashx?proxy"></script>
  <script type="text/javascript">
/* <![CDATA[  */

    window.onload = function() {
    var s = new HelloWorld();
    alert("sync:" + s.greetings());
    s.greetings(function(response) {
    alert("async:" + response.result)

}); } /* ]]> */
  </script>
  </head>
<body>
  <p>This page tests the HelloWorld service with Jayrock.</p>
</body>
```

http://jayrock.berlios.de/

NVelocity

```
<HTML>
<BODY>
Hello $customer.Name!
<table>
#foreach( $mud in $mudsOnSpecial )
#if ( $customer.hasPurchased($mud) )
<tr> <td>
$flogger.getPromo( $mud )
</td> </tr>
#end
#end
</table>
```

http://velocity.apache.org/engine/releases/velocity-1.5/user-guide.html

Spring.NET + Jayrock + NVelocity

- Spring.NET
  - http://www.springframework.net/
- Jayrock
  - http://www.castleproject.org/others/nvelocity/index.html
- NVelocity
  - http://www.castleproject.org/others/nvelocity/index.html

NPOI

- When to use it
  - When you need to read or write **XLS, DOC, or PPT**.
- What it does
  - Exposes an **API** that can called from a .NET application.
- How it works
  - The API reads and writes the **file formats** directly.
- Where does it fit in
  - Data access layer.

- http://npoi.codeplex.com/

http://na11.apachecon.com/talks/19459

husted@apache.org
http://www.slideshare.net/ted.husted